



Universidad
Carlos III de Madrid

DEPARTAMENTO DE INFORMATICA

PROYECTO FIN DE CARRERA

Diseño e implementación de una plataforma para la asistencia en el tratamiento de registros asociados a usuarios.

Autora: Esther Rodríguez Andrés

Tutor: Alejandro Calderón Mateos

Leganés, Junio de 2012

Título: Aplicación de Control y Gestión de Datos representados Gráficamente.

Autor: Esther Rodríguez Andrés

Director: Alejandro Calderón Mateos

EL TRIBUNAL

Presidente/a: _____

Vocal: _____

Secretario/a: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera del
día de de 20 en Leganés, en la Universidad Carlos III de
Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO/A

PRESIDENTE/A

“Si todas las cosas siempre están en su sitio, hasta a oscuras las encontraras”.

Jeremías Rodríguez.

RESUMEN

El proyecto lo que trata es de, obtener de un origen de registros de datos o ficheros logs, una información clara y precisa para que un usuario inexperto sea capaz de visualizarlos de forma intuitiva y sencilla. Obteniendo así la información que necesita de forma directa, sin tener que pelearse con cientos y cientos de registro incomprensibles para él.

Así mismo el administrador del sistema se encargará de crear esta aplicación, que englobe dicho problema. Es decir, leer los datos, almacenarlos cuidadosamente en la base de datos y crear una aplicación que muestre una serie de consultas de forma gráfica para su mejor comprensión.

ABSTRACT

The project is trying to obtain a record source data or log files, clear and accurate information for a novice user is able to visualize in an intuitive and simple. Thus obtaining the information you need directly, without having to fight with hundreds and hundreds of record incomprehensible to him.

Also the system administrator is responsible for creating this application, encompassing the problem. That is, reading the data, store them carefully in the database and create an application that displays a series of queries graphically for easy understanding.

Índice de Contenidos

ÍNDICE DE CONTENIDOS.....	6
ÍNDICE DE TABLAS.....	9
INDICÉ DE ILUSTRACIONES.....	10
INTRODUCCIÓN Y OBJETIVOS	11
1.1 VISIÓN GENERAL	12
1.2 MOTIVACIÓN.....	13
1.3 OBJETIVOS.....	15
1.4 ACRÓNIMOS.....	16
1.5 ESTRUCTURA DEL DOCUMENTO	17
ESTADO DE LA CUESTIÓN	19
2. 1 ESTUDIO PREVIO	20
2.1.1 ESTUDIO DE MERCADO	20
2.1.2 SISTEMA DE CONTROL USUARIOS	22
2.1.3 SISTEMAS GESTORES DE BASES DE DATOS	24
2.1.4 SERVIDORES WEB.....	28
2.1.5 APLICACIONES ESTÁTICAS Y DINÁMICAS	31
2.2 TECNOLOGÍAS APLICADAS.....	33
2.2.1 DESCRIPCIÓN DEL SISTEMA GESTOR DE BASES DE DATOS.....	35
2.2.2 SQL.....	37
2.2.3 DESCRIPCIÓN DE ESTÁNDARES WEB.....	39
2.2.4 HTML.....	40
2.2.5 HOJA DE ESTILOS CSS.....	42
2.2.6 PHP.....	43
2.2.7 JAVASCRIPT	44
2.2.8 LIBRERÍAS JGRAPH	45
ANÁLISIS, DISEÑO E IMPLEMENTACIÓN	47
3.1 ANÁLISIS	48
3.1.1 REQUISITOS DEL SISTEMA	48
REQUISITOS FUNCIONALES	49
REQUISITOS NO FUNCIONALES	59
REQUISITOS DE USABILIDAD:.....	59

3.1.2	CASOS DE USO.....	65
3.1.2.1	DESCRIPCIÓN GRÁFICA	65
3.1.2.2	DESCRIPCIÓN TEXTUAL.....	67
3.2	DISEÑO	74
3.2.1	BASE DE DATOS.....	74
3.2.1.1	MODELO E/R	74
3.2.1.1	MODELO RELACIONAL.....	78
3.2.2	APLICACIÓN	87
3.2.2	APLICACIÓN	87
3.2.2.1	DIAGRAMA DE NAVEGACIÓN	87
3.3	IMPLEMENTACIÓN	89
3.3.1	BASE DE DATOS.....	89
3.3.1.1	CREAR OBJETOS	89
3.3.1.2	CONSULTAS.....	91
3.3.2	APLICACIÓN.....	92
3.3.2.1	DISEÑO VISUAL BASIC	92
3.3.2.2	TRABAJO DE PHP Y GRÁFICOS	97
	PLANIFICACIÓN Y PRESUPUESTO	105
4.1	PLANIFICACIÓN	106
4.1.1	MODELO DE GESTIÓN DEL PROYECTO	106
4.1.2	DIAGRAMA DE GANTT.....	107
4.2	PRESUPUESTO	110
4.2.1	RESUMEN DE HORAS DEDICADAS.....	110
4.2.2	RESUMEN DE PERSONAL.....	111
4.2	RESUMEN DE HARDWARE	112
4.2.4	RESUMEN DE SOFTWARE Y LICENCIAS.....	113
4.2.5	RESUMEN DE MATERIAL FUNGIBLE	114
4.2.6	RESUMEN DEL PRESUPUESTO TOTAL.....	115
4.2.7	PLANTILLA RESUMEN	116
	CONCLUSIONES Y TRABAJOS FUTUROS.....	117
5.1	CONCLUSIONES	118
	PROCESO	118
	PERSONAL.....	119
	PRODUCTO	120

5. 2 TRABAJOS FUTUROS.....	122
BIBLIOGRAFÍA	125
ANEXOS	127
ANEXO 1: MANUAL DE INSTALACIÓN Y USO DEL SERVIDOR	128
1.1 SOBRE ESTE MANUAL.....	128
1.2 INFORMACIÓN GENERAL	128
1.3 INSTALACIÓN	129
ANEXO 2: ACERCA DE LA BASE DE DATOS	131
2.1 CREACIÓN DE TABLAS	131
2.2 DICCIONARIO DE DATOS.....	133
ANEXO 3: ACERCA DE LA APLICACIÓN	135

Índice de Tablas

Tabla 1.....	25
Tabla 10.....	54
Tabla 11.....	54
Tabla 12.....	55
Tabla 13.....	55
Tabla 14.....	56
Tabla 15.....	56
Tabla 16.....	57
Tabla 17.....	57
Tabla 18.....	58
Tabla 19.....	59
Tabla 20.....	59
Tabla 22.....	60
Tabla 23.....	61
Tabla 24.....	61
Tabla 25.....	62
Tabla 26.....	62
Tabla 27.....	63
Tabla 28.....	63
Tabla 29.....	64
Tabla 30.....	64
Tabla 31.....	68
Tabla 32.....	68
Tabla 33.....	69
Tabla 34.....	69
Tabla 35.....	70
Tabla 36.....	70
Tabla 37.....	71
Tabla 38.....	71
Tabla 39.....	72
Tabla 40.....	72
Tabla 41.....	73
Tabla 42.....	111
Tabla 43.....	112
Tabla 44.....	113
Tabla 45.....	114
Tabla 46.....	115
Tabla 5.....	51
Tabla 6.....	52
Tabla 7.....	52
Tabla 8.....	53
Tabla 9.....	53
Tabla2.....	50
Tabla3.....	50
Tabla4.....	51

Indicé de Ilustraciones

ILUSTRACION 1:	44
ILUSTRACIÓN 2:	72
ILUSTRACIÓN 3:	73
ILUSTRACIÓN 4:	83
ILUSTRACIÓN 5:	102
ILUSTRACIÓN 6:	103
ILUSTRACIÓN 8:	103
ILUSTRACIÓN 9:	104
ILUSTRACIÓN 10:	107
ILUSTRACIÓN 11:	108
ILUSTRACIÓN 12:	109
ILUSTRACIÓN 13:	110
ILUSTRACIÓN 14:	112
ILUSTRACIÓN 15:	118
ILUSTRACIÓN 16:	141
ILUSTRACIÓN 17:	141
ILUSTRACIÓN 18:	141
ILUSTRACIÓN 19:	142
ILUSTRACIÓN 20:	142
ILUSTRACIÓN 21:	142
ILUSTRACIÓN 22:	144
ILUSTRACIÓN 23:	145
ILUSTRACIÓN 24:	145
ILUSTRACIÓN 25:	146
ILUSTRACIÓN 26:	147
ILUSTRACIÓN 27:	148

Capítulo 1

Introducción y objetivos

Esta sección tiene como objetivo principal el de presentar una visión general sobre el proyecto realizado, así como indicar las motivaciones y metas por las que se ha decidido realizar este proyecto. Para finalizar con este punto se mostrará una lista de los acrónimos utilizados en el documento con sus correspondientes definiciones para ayudar a la comprensión del contenido.

1.1 Visión general

Por una parte la visión general de este proyecto se basa en la necesidad de realizar una aplicación, que pueda tratar distintos ficheros de datos que se derivan de distintos servidores y representarlo visualmente.

Estos datos están en un formato de difícil comprensión para el usuario. Se componen de cientos de líneas que contienen campos con diferente información y de diferente índole sin un sentido claro nada más que para la máquina.

Se presupone que la información que necesita el usuario se encuentra en las cientos de líneas, pero para el se convierte en una tarea casi imposible de realizar. Es aquí donde se crea la necesidad, que más adelante se explicará, de realizar una aplicación que organice esta información y la almacene ordenadamente para poder acceder a ella fácilmente cuando se necesite.

He aquí un ejemplo; se necesita saber sobre qué horas el servidor de la universidad “snoopy” registra mayor actividad. Los responsables del laboratorio poseen cientos y cientos de listados que incluyen esta información, pero les llevaría gran parte de sus jornadas laborales en devolver dicho dato, por lo que se crea la necesidad de tener una aplicación que se ocupe de ello.

El sistema debe ser capaz de dar esta información en pocos segundos y ser totalmente seguro de que esa información es fiable.

Por otra parte, la necesidad de cumplir unos estándares internos de calidad y usabilidad, hacen de este tratamiento ser debidamente diseñado para ser intuitivo y de fácil manejo para todo tipo de usuarios. También debe ser una aplicación abierta a futuras modificaciones, puesto que se crearan nuevas necesidades a lo largo de la vida funcional del producto.

Por ejemplo, en una empresa particular cada fin de mes necesitan obtener un listado genérico de los operarios que llegan tarde, pero en los próximos meses se necesitará además quienes de aquellos han llegado más pronto para un plus de puntualidad. Esta sería una nueva necesidad a cubrir y que la aplicación debe estar preparada para ella.

Cabe aquí destacar, que la información final mostrada al usuario será en formato grafico, es decir, en un eje de abscisas. Así se podrá ver de forma más clara y tener una visión de los sucesos anteriores y posteriores al caso en cuestión.

1.2 Motivación

La clara motivación de este proyecto es cubrir las necesidades planteadas por el cliente, y no habiendo encontrado la aplicación idónea se crea la necesidad de crear una propia.

Con este proyecto se pretende crear una herramienta que sea capaz de almacenar y visualizar los datos recogidos por otra aplicación de manera que facilite la gestión a los usuarios.

Para ser más esquemáticos:

- Organización de los datos obtenidos de diferentes orígenes. Datos obtenidos en primera instancia de máquinas informáticas, que devuelven los datos en un formato incomprensible para el usuario.
- Obtención de respuestas claras y concisas que representen al usuario una visión directa de lo que quiere. La elección de la solución será en una visión grafica, que ayude al usuario a identificarla de forma rápida y sencilla.
- Motivación futura adquirida por el administrador bajo su experiencia y conocimientos de nuevas necesidades por parte del cliente.

En conclusión, se crea esta aplicación por la motivación de poseer una herramienta que englobe dos características. Capacidad de tratamiento de datos en formato lógico y capacidad de representar dichos datos de una forma clara en pantalla que haga fácilmente su comprensión.

Hoy en día existen numerosas aplicaciones que realizan operaciones parecidas, pero no llegan a englobar una solución total. Sin embargo, las nuevas tecnologías y avances técnicos hacen que esto sea por un periodo corto de tiempo.

Así también y en contrapunto se anota el avance en las tecnologías SAP que hoy en día se empiezan a integrar en las empresas privadas y públicas para la gestión conjunta de las operaciones internas de ellas mismas. Estas tecnologías desarrollan un control de la producción, la contabilidad y la organización además de otras áreas.

Son potentes herramientas que dan una solución óptima y rápida a las posibles necesidades del mercado y el desarrollo de la actividad.

Se debe indicar que el proyecto se centra en un área muy pequeña de desarrollo, por lo que no puede ser comparada con la anterior mención de las tecnologías SAP, pero sí que intenta tomar como referencia este tipo de trabajo. Producción rápida, organizada y sencilla.

Pongamos un ejemplo; el director adjunto de una empresa desea una relación de los trabajadores que llegaron a cierta hora y se fueron a otra, para investigar si alguno fue culpable de un material sustraído. Es muy posible que esta casuística tan concreta no sea capaz de englobarla una gran aplicación robusta como son las tecnologías SAP. Será el departamento informático interno quien deberá generar una aplicación para ello. Es aquí donde se demuestra la verdadera motivación del proyecto realizado. Será é quien bajo ciertos datos de fichajes dirima esta información y se la muestre al director de una forma rápida y eficaz, con la absoluta confianza de que los datos devueltos son exactos.

La verdadera motivación es una solución barata y eficaz para la empresa que así lo solicita.

1.3 Objetivos

El objetivo principal de este proyecto es desarrollar una tecnología actual que ayude a realizar nuevas tareas y tomar decisiones tras los resultados obtenidos.

Estudiando las posibles herramientas disponibles para llevar a cabo el almacenamiento y representación de los datos, se han propuesto una serie de objetivos más concretos para la realización del prototipo de una aplicación que hará posible el objetivo principal del proyecto:

- Diseño y desarrollo de una aplicación que muestre de forma clara la información en tiempo real de una situación concreta.
- Estudio de las posibilidades que ofrece las nuevas tecnologías para insertar estas en nuestro proyecto, haciendo las más potentes y amigables de manejar.
- Búsqueda de soluciones a las limitaciones de desarrollo de las propias herramientas.
- Implantación de una base de datos estable que sea capaz de almacenar los datos de forma correcta y ordenada además de ayudar a un trato fácil de la operación de mantenimiento de la propia base de datos.
- Realización de gráficos usando los datos almacenados en la base de datos.

1.4 Acrónimos

API: Application Programming Interface

CSS: Cascading Style Sheets

FTP: File Transfer Protocol

HTML: HyperText Markup Language

HTTP: HyperText Transfer Protocol

PHP: Hypertext Pre-Processor

SCGI: Simple Common Gateway Interface

SGBD: Sistema Gestor de Bases de Datos

SRC: Source

SSL: Secure Sockets Layer

SQL: Structured Query Language

URL: Uniform Resource Protocol

W3C: World Wide Web Consortium

XML: eXtensible Markup Language

1.5 Estructura del documento

Este apartado está destinado a explicar brevemente los puntos a desarrollar en los capítulos de este documento.

Capítulo 1:

Se pretende dar una visión global del proyecto realizado, los objetivos establecidos, y los acrónimos usados en el resto del documento para la una mayor comprensión.

Capítulo 2:

En el Estado de la Cuestión se tratarán temas relacionados con el control de usuarios y el trabajo con los ficheros de datos a través de servidores Apache y tecnologías estudiadas para hacer posible el desarrollo del proyecto.

Capítulo 3:

Este capítulo muestra la herramienta y técnicas desarrolladas en la fase de análisis y de diseño del proyecto. Además se mostrarán los aspectos más destacados llevados a cabo en la fase de implementación.

Capítulo 4:

En Planificación y Presupuesto se analizará el Diagrama de Gantt, método utilizado para la planificación, y se desglosarán los costes que han hecho posible el desarrollo del proyecto.

Capítulo 5:

Conclusiones y Trabajos Futuros, es un capítulo subjetivo acerca de los objetivos conseguidos con la elaboración de este proyecto así como las posibles mejoras a realizar en los trabajos futuros. Además, este apartado incluye la Bibliografía utilizada.

Capítulo 6:

Este capítulo contiene los anexos realizados para, en caso de ser necesario, llegar a un mejor rendimiento del funcionamiento del servidor utilizado, de la base de datos, y de la aplicación web.

Capítulo 2

Estado de la Cuestión

En este apartado se estudian y analizan los sistemas gestores de bases de datos y los servidores web así como los estándares más utilizados en la actualidad dependiendo de la naturaleza de su contenido (estático y dinámico) y de las estructuras que presentan, con esto se pretende explicar los motivos de las decisiones tomadas acerca de las tecnologías aplicadas en el desarrollo del proyecto. Además se hablará brevemente de los lenguajes que se han utilizado en la implementación.

2.1 Estudio Previo

2.1.1 Estudio de Mercado

En el mercado de hoy en día, lo más importante es la productividad, la cual normalmente derivará en el beneficio. Esta productividad vendrá precedida por el trabajo realizado. Cuando este trabajo es procedente de los sub-trabajos que realizan los distintos operarios se necesita tener un control sobre ellos.

Así el control de presencia de estos operarios, como las operaciones que realizan es muy importante. Hoy en día se están estudiando numerosas tecnologías para este aspecto y se apuesta cada vez más por ellas. Son cada vez más sofisticadas y orientadas al control individual del operario. Como ejemplo, el control ocular o dactilar.

Unido a esto, cabe destacar, que la información resultante se necesita de forma clara, precisa y rápida. Es por ello que las tecnologías y aplicaciones que sean capaces de ofrecer esto último serán las elegidas para ser las más comercializadas, con lo que esto significa para una negocio.

Imagine el lector como puede ayudar una aplicación como esta a un operario o usuario que le sea asignada una de las tareas anteriores. No es lo mismo apretar un botón y obtener la información claramente que obtener numerosa información y tener que organizarla para obtener una pocas conclusiones de forma tediosa y lenta.

Lo apuntando anteriormente es una posible situación de mercado, pero imaginemos el beneficio que supone una aplicación que sea capaz de obtener graficas de producción o de mercado. Que producto se consume más, a que días o a qué horas, y quienes. Y así multitud de posibles situaciones que pueden ayudar a tomar decisiones beneficiosas para la buena marcha de un negocio o profesión.

En conclusión, se puede afirmar que se están realizando en la actualidad, numerosas investigaciones para mejorar la tecnología de los sistemas de control y de administración de la información, pero todavía no existen aplicaciones para representar de forma gráfica los datos y ayudar al usuario a la interpretación de los mismos.

2.1.2 Sistema de Control Usuarios

El sistema de control de usuarios consiste en una aplicación que tras una serie de almacenamientos de datos y consultas a ellos obtiene una determinada información concretada con anterioridad. Esta información servirá para mostrar en pantalla en formato grafico los resultados consultados. Esta gráfica ayudará a ver la información de forma más clara y rápida para poder tomar las decisiones pertinentes.

El sistema consta de una base de datos, una aplicación creada para el proyecto, una interfaz web y un servidor Apache capaz de gestionar todo lo anterior. La base de datos alojará la información necesaria, la aplicación será la encargada de mostrar formularios o pantallas para que el usuario administrador y no administrador elijan su proceso de acción. Y por último el servidor Apache relacionará la base de datos con la aplicación y subirá la página web al Servidor Web que incluye el propio servidor.

Este sistema es creado para la ocasión. Es decir, no se basa en antiguos sistemas o nuevas tecnologías. Se basa en la necesidad existente para el control de los usuarios de distintos sistemas. Así se podrá instalar en distintos lugares.

Se podrá asociar a la entrada/salida de profesores en el departamento, a la productividad de los operarios de un taller, a la posible suplantación de personas en lugares poco controlados como la acción de mercado sobre un producto.

Este tipo de sistemas se caracteriza por la sencillez de desarrollarlas e implantarlas. El sistema de control de usuarios consta de numerosas virtudes, algunas a destacar son:

- **Uso eficiente de energía:** Son sistemas que necesitan de un consumo mínimo de energía para su funcionamiento, lo que permite una mayor autonomía.
- **Escalabilidad:** Las redes pueden ser ampliadas en cualquier momento de una manera fácil, pues si se añaden equipos de control al sistema, el resto lo reconocen automáticamente y sólo se tendrían que actualizar las tablas de rutas.
- **Fácil implantación:** Simplemente será necesario un ordenador que sea capaz de alojar la información del sistema recogida por el aparato instalado. Bien un sistema de digitalización dactilar o sistema de tarjetas por ejemplo. Esto será a gusto de propietario del sistema. La recogida de la información es inherente a el sistema electrónico que realice la captación de las tomas de usuario.
- **Bajo coste:** Este tipo de sistemas no necesitan de una gran máquina de procesos, cualquier PC de rutina es suficiente. El sistema software instalado en cada uno de los equipos instalados es mínimo, puesto que se ha apostado por software libre y propio del desarrollador del proyecto.

Gracias a todas estas características se permite implantar este sistema en numerosos escenarios y para diversas acciones, como anteriormente se ha descrito.

2.1.3 Sistemas gestores de bases de datos

Los sistemas de gestión de bases de datos son un tipo de software específico dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones web que la utilizan. Estos sistemas se encargan de manejar los datos que se van a utilizar en las aplicaciones de una manera clara, sencilla y ordenada.

Los objetivos de utilizar un SGBD son abstraer la información, conseguir mayor seguridad, facilidad en el manejo de transacciones y obtener una independencia y consistencia de los datos. De este modo se ahorra al usuario detalles acerca del almacenamiento físico de los datos y se garantiza que la información se encuentra segura para todos los usuarios y se respetan los permisos asignados a cada uno.

Además de ventajas para los usuarios, los SGBD facilitan la manipulación de grandes volúmenes de datos y la programación para los desarrolladores, por tanto, bajan drásticamente los tiempos de desarrollo y aumentan la calidad del sistema. Como inconvenientes solo destacaremos la necesidad de disponer de administradores de la base de datos, lo que puede incrementar los costes de operación así como un coste adicional para el hardware necesario, ya que los requisitos de hardware para correr un SGBD, por lo general, son relativamente altos.

De los posibles SGBD libres disponibles en el mercado se han analizado las características de los más utilizados, MySQL, SQL Server y Oracle. A continuación se muestra una tabla esquemática con ventajas y desventajas que han ayudado a la elección de MySQL para el proyecto.

	MySQL ₁	SQLServer ₂	Oracle ₃
S.O	Multiplataforma	Microsoft Windows	Multiplataforma
Licencia	GPL	Microsoft EULA	Privada
Relacional	SI	SI	SI
Multihilo	SI	SI	SI
Multiusuario	SI	SI	SI
Disparadores	SI	SI	SI
Claves ajenas	SI	SI	SI
Vistas actualizables	SI	SI	SI
Procedimientos almacenados	SI	SI	SI
Escalabilidad	NO	SI	SI
Estabilidad	NO	SI	SI
Conectividad segura	SI	SI	SI

Tabla 1: Comparativa de sistemas gestores de bases de datos

¹ www.mysql.com/

² www.microsoft.com/sqlserver/2008/en/us/

³ www.oracle.com/es/index.html

Como conclusiones podemos decir que MySQL es ligero y relativamente rápido, pero algo escaso de características, mientras que SQL Server es más robusto y aunque tiene un buen rendimiento, es más lento que MySQL. Por otro lado para utilizar SQL Server hay que pagar las licencias de Microsoft, en cambio, MySQL es gratuito, ya que se ofrece bajo la GNU GPL (*GNU General Public License*) para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privados, deben comprar una licencia específica que les permita este uso.

SQL Server es capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Sus principales ventajas son la escalabilidad, estabilidad y seguridad, además, permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red solo acceden a la información.

La tecnología Oracle se encuentra prácticamente en todas las industrias alrededor del mundo, es el proveedor mundial líder de software para administración de información, y la segunda empresa de software. Oracle está considerado uno de los sistemas de bases de datos más completos. A pesar de todo esto el coste de las licencias para su uso es elevado, este ha sido el principal motivo para no utilizar este SGBD.

En la aplicación web a desarrollar hay baja concurrencia en la modificación de los datos pero el entorno es intensivo en lecturas, lo que hace que MySQL sea ideal para cumplir con nuestros requisitos, además de que al tener las licencias gratuitas el acceso a la plataforma es más fácil.

2.1.4 Servidores Web

Un servidor web es un programa diseñado para transferir hipertextos, páginas web o páginas HTML, implementando el protocolo HTTP. El servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (navegador web) y responde a estas peticiones adecuadamente, mediante una página web que se mostrará en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Además de la transferencia de código HTML, los servidores web pueden entregar aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- **Aplicaciones en el lado del cliente:** El cliente web es el encargado de ejecutarlas en la máquina del usuario. El servidor proporciona el código de las aplicaciones al cliente y éste las ejecuta a través del navegador. Para esto es necesario que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones.

- **Aplicaciones en el lado del servidor:** El servidor web ejecuta la aplicación y una vez ejecutada se genera un código HTML, el servidor toma el código recién creado y lo envía al cliente por medio del protocolo HTTP.

El término servidor también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizarlos. Este uso dual puede llevar a confusión, un servidor web podría referirse a la máquina que almacena y maneja los sitios web y también podría aludir al software que funciona en la máquina y controla la entrega de los componentes de las páginas web como respuesta a las peticiones de los navegadores.

Existen empresas dedicadas al "*hosting*", es un negocio que consiste en alojar, servir y mantener archivos para uno o más sitios web. La mayoría de los servicios de *hosting* ofrecen conexiones que para una persona individual resultarían muy costosas. Usar este servicio permite que muchas compañías compartan el coste de una conexión rápida a internet para el acceso a los archivos de sus sitios web.

Hay una amplia variedad de servicios de *hosting*. El más básico es el de archivos donde se pueden alojar las páginas de los sitios web y otros archivos vía ftp o una interfaz web. Normalmente no se requiere de grandes inversiones de dinero para alojar las páginas web en servidores, existen algunos gratuitos que aunque tienen límites muy grandes de espacio y de tráfico, comparado con el de pago, son suficientes para pequeñas aplicaciones.

Otra opción es instalar un servidor web en nuestro ordenador para poder montar una página web propia sin necesidad de contratar *hosting*. El problema de usar nuestro ordenador como servidor web local es que se debería tener encendido permanentemente para que esté accesible de forma continua, como el resto de sitios web, y el coste de recursos y electricidad sería importante.

Algunos de los servidores web más importantes son Apache⁴ y Cherokee⁵. Apache es el servidor con más años de experiencia en el mundo de los desarrolladores web, produciendo todo tipo de soluciones de software para aplicaciones muy diversas. En los últimos años se está oyendo hablar de Cherokee como una alternativa viable a Apache, más rápida y fácil de usar.

Apache es un servidor web destacado por jugar un papel importante en el crecimiento inicial de la *World Wide Web*. Es desarrollado y mantenido por una comunidad abierta de desarrolladores. La aplicación está disponible para una amplia variedad de sistemas operativos, se caracteriza por ser código abierto.

⁴ www.apache.org

⁵ www.cherokee-project.com

Cherokee es muy rápido, flexible y fácil de configurar. Es compatible con las tecnologías más usadas hoy en día: FastCGI, SCGI, uWSGI, SSI TLS y conexiones cifradas SSL, hosts virtuales, la autenticación, la codificación de ventanas, de equilibrio de carga, streaming de video...etc.

Apache se utiliza principalmente para servir contenido estático y dinámico de páginas web en la *World Wide Web*. Es un servidor web usado para muchas tareas, donde el contenido necesita ser puesto a disposición de una manera segura y confiable. Un ejemplo es el intercambio de archivos desde un ordenador personal a través de internet, un usuario que tiene Apache instalado en su escritorio puede colocar archivos arbitrarios en la raíz de documentos del servidor y pueden ser compartidos.

Aunque el objetivo principal del servidor no es ser el más rápido, Apache tiene un rendimiento comparable a otros servidores web de "alto rendimiento". Se compromete con el funcionamiento de manera que reduce la latencia y aumenta el rendimiento en los momentos de más peticiones, así se asegura un tratamiento constante y fiable de peticiones dentro de periodos razonables.

2.1.5 Aplicaciones estáticas y dinámicas

Las aplicaciones web se dividen en dos grandes grupos; webs estáticas y webs dinámicas. A continuación, se analizarán las principales diferencias entre ambas, lo que permitirá entender la elección tomada para realizar esta aplicación web.

Una web estática presenta las siguientes características:

- Ausencia de movimiento y funcionalidades
- Absoluta opacidad a los deseos o búsquedas del visitante a la página
- Realizadas en XHTML o HTML
- Para modificar los contenidos de la página, es imprescindible acceder al servidor donde está alojada la página
- El usuario no tiene posibilidades de seleccionar, modificar u ordenar los contenidos o el diseño de la página a su gusto
- El proceso de actualización es lento, tedioso y esencialmente manual
- No se pueden utilizar funcionalidades tales como bases de datos, foros...

Por el contrario, una web dinámica tiene las siguientes características:

- Gran número de posibilidades en su diseño y desarrollo.
- El visitante puede alterar el diseño, contenidos o presentación de la página a su gusto.
- En su realización se pueden usar diferentes lenguajes y técnicas de programación.
- El proceso de actualización es sencillo, sin necesidad de entrar en el servidor.
- Permite un gran número de funcionalidades como bases de datos, foros... etc.
- Puede realizarse íntegramente con software libre.
- Cuenta con un gran número de soluciones prediseñadas de libre disposición

Como idea principal podemos decir que si solo se requiere informar al público visitante sobre productos o servicios, una web estática sería la elección correcta, pero si se desea interactuar con nuestra web entonces se necesitaría una web dinámica.

Una web dinámica es mucho más costosa que una web estática en cuanto a la programación de la misma. No obstante, una web estática puede tener los mismos o más costes que una web dinámica, ya que, en algunas ocasiones, la complejidad reside en el grado de sofisticación del diseño.

No es suficiente tener un buen diseño para obtener una mayor rentabilidad del sitio web, también es necesario tener un contenido de calidad y actualizado.

Hay que unir diseño y contenido y esto sólo se puede obtener con una web dinámica.

Como conclusión se puede decir que los dos tipos de web pueden ser una buena opción, depende de los objetivos planteados. Por tanto, nuestra aplicación será dinámica para poder cumplir los objetivos que nos hemos propuesto alcanzar.

2.2 Tecnologías aplicadas

Para la solución de este proyecto se ha utilizado el servidor independiente de software libre 'WAMPP'. Este servidor consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y PERL. El programa está liberado bajo la licencia GNU y actúa como servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Este servidor ha sido el elegido por su fácil instalación y porque permite testear el trabajo en el ordenador personal sin ningún acceso a internet.

En el proceso de construcción de la base de datos podemos distinguir entre dos fases; fase de diseño y de implementación. Para la fase de diseño se han utilizado los modelos de datos siguientes; el modelo E/R (Chen P., 1976) y el modelo relacional (Codd E., 1970). Por otro lado, para las fases de construcción, carga de la base de datos y recuperación de los datos en la aplicación, el lenguaje utilizado ha sido SQL.

La realización del modelo E/R o Entidad Interrelación ha servido para aclarar ideas sobre la situación y las relaciones de los conceptos antes de pasar a la realización definitiva del diseño lógico de la base de datos, además de que supone un nivel mayor de abstracción y por tanto, una forma más cómoda de trabajar al simplificar el problema. Representado en el apartado (3.2.1.1) de este documento.

El modelo relacional es el esquema lógico de la fase de diseño de la base de datos. Este modelo es propio del tipo de base de datos que se va a implementar, tipo relacional. El diseño lógico debe ser de este tipo para obtener así una implementación directa desde el esquema diseñado. Representado en el apartado (3.2.1.2) de este documento.

En las tareas de construcción y carga de la base de datos, así como la recuperación de información de la misma, nos hemos centrado en un único lenguaje, teniendo en cuenta que debía ser compatible con el sistema gestor de bases de datos, en este caso es MySQL. La tecnología elegida es el lenguaje declarativo de bases de datos de alto nivel SQL. El lenguaje fue estandarizado por ANSI (Instituto Nacional Estadounidense de Estándares) y posteriormente aceptado por ISO (Organización Internacional para la Estandarización), este último es el organismo de estandarización internacional más importante.

⁶ <http://www.wampserver.com>

2.2.1 Descripción del Sistema Gestor de Bases de Datos

MySQL es un Sistema Gestor de Bases de Datos relacionales, multiusuario, multihilo y desarrollado en ANSI C y C++. Por un lado se ofrece bajo la GNU/GPL para cualquier uso compatible con ésta, sin embargo, aquellas empresas que quieran incorporarlo para fines privados deben comprar a MySQL AB una licencia específica que les permita este uso.

MySQL es muy utilizado en plataformas Linux/Windows-Apache-MySQLPHP/Perl/Python. Su popularidad como aplicación Web está muy ligada a PHP, que a menudo aparece como combinación con MySQL, ya que gracias a su rapidez es el gestor ideal para este tipo de aplicaciones. A continuación se listan algunas de sus características más destacadas:

- Soporte a multiplataforma
- Un amplio subconjunto de ANSI SQL 99, y varias extensiones
- Procedimientos almacenados
- Cursores
- Disparadores ("*trigger*")
- Soporte a Varchar
- Vistas actualizables
- Motores de almacenamiento independientes (MyISAM para lecturas rápidas, InnoDB para transacciones e integridad referencial)
- Soporte para SSL
- Permite sentencias SELECT anidadas
- Soporte completo para Unicode

En el paquete de aplicaciones elegido para desarrollar el proyecto (XAMPP) se incluyen módulos como Open SSL o phpMyAdmin. Esta última es una herramienta escrita en PHP que actúa de interfaz con el sistema gestor de bases de datos a través de aplicaciones web, que permite ejecutar sentencias SQL en el navegador web y realizar operaciones sobre bases de datos, administración, tablas e importación y exportación de datos.

Durante todo el desarrollo del proyecto se ha utilizado esta aplicación para interactuar con la base de datos. En la siguiente figura se muestra la apariencia de la herramienta con la base de datos del proyecto en pantalla:

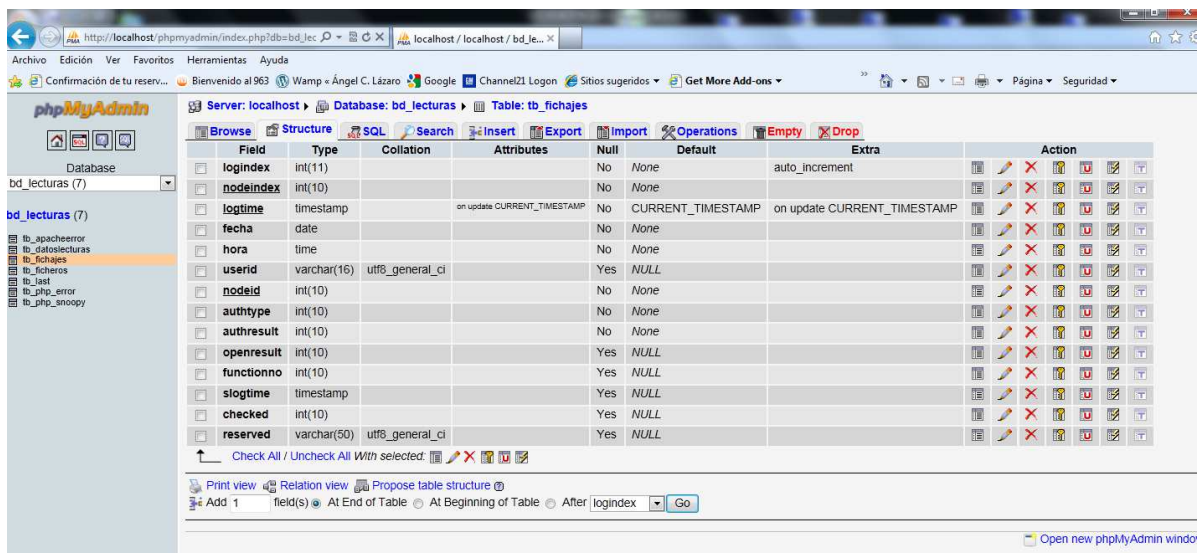


Ilustración 1: Apariencia de la herramienta de datos.

Este paquete de aplicaciones ha sido elegido por ser el más adecuado y adaptarse a las necesidades planteadas. Este servidor contiene todas las herramientas necesarias para la realización de las tareas que comprenden este proyecto.

Por comodidad a la hora de trabajar y para evitar imprevistos, se ha optado por una versión portable del servidor. Esta versión permite transportar toda la aplicación de manera sencilla en una memoria portátil USB, de este modo podemos trabajar en diferentes máquinas lo que facilita el trabajo del desarrollador.

El manual completo de instalación se encuentra en los anexos a este documento.

2.2.2 SQL

El lenguaje de consulta estructurado o SQL es un lenguaje declarativo de acceso a bases de datos relacionales que permite la especificación de diversos tipos de operaciones sobre éstas. Actualmente es uno de los lenguajes de bases de datos más usado, tanto es así que SQL es el estándar que deben seguir los sistemas gestores de bases de datos relacionales.

Está considerado un lenguaje de alto nivel gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, lo que permite una alta productividad en codificación y orientación a objetos. De este modo una única sentencia podría equivaler a uno o más programas que se utilicen en un lenguaje de bajo nivel.

Está compuesto por dos grandes sub-lenguajes con una temática bien diferenciada. Para la declaración de las estructuras y objetos de la base de datos se utilizará el Lenguaje de Definición de Datos (LDD) y para las fases de carga y consulta, SQL dispone del Lenguaje de Manipulación de Datos (LMD).

Para el lenguaje LDD existen cuatro operaciones básicas, todas ellas pueden actuar sobre una tabla, vista, índice, *trigger*, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte:

- **CREATE:** Este comando crea un objeto dentro de la base de datos.
- **ALTER:** Este comando permite modificar la estructura de un objeto.
- **DROP:** Este comando elimina un objeto de la base de datos. Se puede combinar con la sentencia ALTER.
- **TRUNCATE:** Este comando trunca todo el contenido de una tabla. La ventaja sobre el comando DROP, es que si se quiere borrar todo el contenido de la tabla, es mucho más rápido, especialmente si la tabla es muy grande. La desventaja es que TRUNCATE sólo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula *Where*.

El lenguaje LMD es un lenguaje proporcionado por el sistema de gestión de bases de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, existen cuatro operaciones básicas:

- **INSERT:** Agrega uno o más registros a una, y sólo una, tabla en una base de datos relacional. Las cantidades de columnas y valores deben ser iguales. Si una columna no se especifica le será asignado el valor por misión.
- **UPDATE:** Esta sentencia es utilizada para modificar los valores de un conjunto de registros existentes en una tabla
- **DELETE:** Borra uno o más registros existentes en una tabla
- **SELECT:** Selecciona uno o más registros de una tabla de la base de datos

2.2.3 Descripción de estándares web

Los estándares web es un término muy general utilizado para referirse a estándares y otras especificaciones que definen y describen aspectos de *la World Wide Consortium (W3C)*. Cuando se describe que un sitio web cumple con ciertos estándares web, usualmente quiere decir que la página tiene partes de código HTML, CSS y Java Script válido o casi válido. La parte HTML debe cumplir también ciertas guías de accesibilidad y semántica.

Un sitio basado en estándares web mostrará una mayor consistencia visual.

Gracias al uso de XHTML para el contenido y CSS para la apariencia, se puede transformar rápidamente un sitio web. Los documentos que separan apariencia de contenido usan menos código, además, CSS permite conseguir efectos que antes requerían el uso de JavaScript e imágenes, por lo que los sitios basados en estándares utilizan menos ancho de banda y se muestran más rápido a los usuarios.

Utilizando los estándares web, el sitio web es más fácil de mantener y actualizar, el código es más simple y más accesible, permitiendo a personas con discapacidades utilizar su contenido.

2.2.4 HTML

HTML son las siglas de *HyperText Markup Language* (Lenguaje de Marcas de Hipertexto), es el utilizado para la construcción de páginas web. Su uso está destinado a describir la estructura y contenido de los sitios web en forma de texto, así como para complementar el texto con objetos tales como imágenes.

Este lenguaje se construye mediante etiquetas que delimitan las secciones de la web y alojan la información. Normalmente suele estar implementado junto con hojas de estilo llamadas CSS que ayudan a definir los aspectos visuales de la web. HTML puede incluir scripts (JavaScript, PHP) que afectan al comportamiento de navegadores web y otros procesadores de HTML.

Puede ser creado y editado por cualquier editor de textos básico, como Gedit en Linux o Notepad++ en Windows, simplemente hay que tener en cuenta que la extensión del documento ha de ser *.htm o *.html. Dentro de estos archivos se implementará el código HTML el cual consta de varias etiquetas básicas:

- **<head>**: Define la cabecera del documento HTML, esta cabecera suele contener información sobre el documento que no se muestra directamente al usuario, como puede ser el título de la ventana del navegador.
- **<title>**: Define el título de la página que aparecerá encima de la ventana del sitio web.
- **<html>**: Define el inicio de un documento HTML, de esta forma se le indica al navegador que el texto que le sigue debe ser interpretado como código HTML.
- **<script>**: Introduce un script en una web, o se llama a uno mediante el atributo "src= 'url del script'".
- **<link>**: Para vincular el sitio a hojas de estilo o iconos.

- **<style>**: Coloca el sitio interno de la página, ya sea usando CSS u otros lenguajes similares.
- **<a>**: Para crear un enlace es necesario utilizar esta etiqueta de ancla junto al atributo href, que establecerá la dirección URL a la que apunta el enlace.
- **<body>**: Define el cuerpo o contenido principal del documento. Esta es la parte del documento que se muestra en el navegador. Dentro del cuerpo se pueden definir numerosas etiquetas, las más utilizadas son:
 - **<p>**: Define un párrafo de texto en la web.
 - **
**: Introduce un salto de línea
 - **<h1>**: Indica que se va a escribir una cabecera para una sección, el número indica el tamaño de la fuente.
 - **<td>**: Define tablas en la página.
 - ****, ****: Definen listas de elementos, ya sean ordenadas o no.

2.2.5 Hoja de estilos CSS

CSS o lo que es lo mismo, hojas de estilo en cascada, es un lenguaje utilizado para definir la presentación de un documento escrito en HTML o XML. El W3C es el encargado de formular las especificaciones de las hojas de estilo que servirán de estándar para los navegadores.

La idea que se mantiene para el desarrollo de las hojas de estilo es separar la estructura del documento de su presentación, lo que hace más fácil la Codificación de los estilos en caso de ser necesario.

Cuando se utiliza CSS, las etiquetas no deben proporcionar información acerca de la visualización del documento, sino marcar la estructura. La información del estilo, separada en una hoja de estilo, especifica como se ha de mostrar una etiqueta; color, fuente, alineación del texto o tamaño.

La introducción de CSS ha permitido en muchos casos reemplazar el uso de tablas, que era la técnica que se utilizaba antes para conseguir el mismo fin. Sin embargo, CSS todavía no permite la misma versatilidad pues las diferencias entre los distintos navegadores dificultan la tarea. Se espera que futuros desarrollos en CSS3 resuelvan esta deficiencia y hagan de CSS un lenguaje más apto para describir la estructura espacial de una página. Como ventajas cabe destacar:

- Los navegadores permiten a los usuarios especificar la hoja de estilo local que será aplicada en un sitio web, con lo que se obtiene una mayor accesibilidad.
- Control centralizado de la presentación de un sitio web completo con lo que se agiliza la actuación del mismo de forma considerable.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

2.2.6 PHP

El lenguaje PHP es un lenguaje de programación diseñado y orientado a la creación de páginas dinámicas. Es utilizado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otro tipo de aplicaciones en las que se encuentran aquellas con interfaz gráfica.

La mayor fuerza de PHP reside en que es un lenguaje preparado para soportar accesos a muchos tipos de bases de datos como Oracle, ODBC, DB2, SQLServer...etc., y enfocando nuestro caso, MySQL. Lo que hace diferente a PHP es que el código que se deba ejecutar se ejecuta siempre en el servidor, de este modo el cliente solo recibe los resultados de la ejecución y le es imposible acceder al código que generó la página.

Las conexiones de PHP a bases de datos son enlaces SQL que no se cierran cuando termina la ejecución del script. El comportamiento de estas conexiones consiste en que al invocar una conexión, PHP comprueba si ya existe una conexión de este mismo tipo o es una conexión nueva. En el caso de que exista se procede a su uso y en caso que no exista la conexión se crea. Dos conexiones se consideran iguales cuando están realizadas al mismo servidor y con un mismo usuario y contraseña.

El código PHP se encuentra inmerso entre el código HTML y Java Script distinguido entre etiquetas:

```
<? Php  
código...  
..... ?>
```

2.2.7 JavaScript

JavaScript es un lenguaje basado en objetos y guiado por eventos, utilizado para acceder a objetos en aplicaciones. Se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. Es un lenguaje influenciado de múltiples lenguajes y fue diseñado con el objetivo de tener una sintaxis similar a la de Java, aunque más sencilla para facilitar el uso a programadores principiantes. Todos los navegadores actuales interpretan el código de JavaScript integrado dentro de las páginas web.

Entre las acciones típicas que se pueden realizar en JavaScript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien color o cualquier otro dinamismo. Por el otro, JavaScript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas o tablas de cálculo.

Este lenguaje se puede incluir en cualquier documento y es compatible con cualquier sistema operativo. La mejor manera es incluir JavaScript como un archivo externo, tanto por cuestiones de accesibilidad como por la velocidad en la navegación, este archivo será del tipo *.js y para utilizarlo se debe escribir en el documento HTML:

```
<script type="text/javascript" src="[url del fichero js]"></script>
```

También es posible incrustar el código en el documento con la etiqueta <script>:

```
<script type="text/javascript"><!--// código JavaScript--></script>
```

2.2.8 Librerías JGraph

Al utilizar las herramientas de Google Maps nos hemos encontrado alguna dificultad para realizar los objetivos de este proyecto. La idea principal era representar, en las pestañas de los mapas de Google, gráficos estadísticos sobre los datos almacenados, estos gráficos también se iban a realizar con herramientas de Google puesto que resultaban más vistosos y óptimos, pero este trabajo ha sido imposible llevarlo a cabo pues las pestañas solo admiten código HTML, por tanto se limitaba el contenido de las pestañas a imágenes, texto, tablas o botones.

Como alternativa se han utilizado las librerías JGraph, estas librerías gratuitas sirven para crear imágenes con todo tipo de gráficas, dinámicamente desde páginas PHP. Incluye una serie de clases en código orientado a objetos que ofrecen una solución muy interesante para la generación de las imágenes con las gráficas, de manera que solo hay que preocuparse de cargar los datos a representar y escoger el tipo de gráfica que se desea visualizar.

El modo de trabajo para usar esta librería es muy simple, se trata de crear una imagen con la etiqueta `` de HTML, en cuyo atributo `src` se coloca la ruta hacia el script PHP que se encargará de generar la gráfica, resultando muy fácil integrar el gráfico en el diseño web.

En el archivo PHP que generará la gráfica tendremos que incluir las librerías apropiadas para el tipo de gráfica que se va a utilizar, también habrá que instanciar el objeto JGraph correspondiente, cargar los datos a visualizar y llamar a los métodos adecuados para mostrar la imagen.

Capítulo 3

Análisis, Diseño e Implementación

En este capítulo se van a desarrollar las diferentes tareas realizadas en las fases de análisis, diseño e implementación diferenciando en cada apartado entre la base de datos, la plataforma web y la aplicación en Visual Basic.

3.1 Análisis

3.1.1 Requisitos del sistema

A continuación, se definen cada uno de los campos de las tablas en las que se describen los requisitos software.

ID: Identificador univoco del requisitos. Debe seguir la siguiente nomenclatura:

- RSF_XX: Requisitos Software Funcionales
- RSNF_XX: Requisitos Software no Funcionales
- RSU_XX: Requisitos Software de Usabilidad

Siendo XX números comprendidos entre 01 y 99 que se incrementarán en una unidad consecutiva con cada nuevo requisito.

Tipo: Pueden ser funcionales, de interfaz, de usabilidad o de mantenimiento.

Patrón(es): Enumera los patrones en los que se ha basado para realizar el requisito. Este campo solamente estará presente en los requisitos funcionales y de usabilidad, si procede.

Descripción: Explicación del requisito.

Importancia: Define la importancia de la funcionalidad que aporta el requisito al proyecto. El dominio de este campo puede tomar los valores "Vital", "Importante" o "Aconsejable".

Justificación: Motivo por el cual se ha decidido incluir el requisito

Criterio de cumplimiento: Método para comprobar que el requisito se ha cumplido

Requisitos Funcionales

ID	RSF_01
Tipo	Funcional
Patrón(es)	ME1 (<i>Hierarchical Organization</i>), BE1 (<i>Collection Center</i>), BE2 (<i>Node as a Single Unit</i>), BN2 (<i>Site Maps</i>), AE1 (<i>User centred structure</i>)
Descripción	Al tratar una cantidad ingente de información se establece un árbol jerárquico con los conceptos extraídos para establecer niveles de importancia y relaciones para poder así clasificarla y representarla de manera lógica.
Importancia	Vital
Justificación	Sin este requisito no se podrían establecer módulos para los distintos apartados y la información estaría reflejada sin ningún orden.
Criterio de cumplimiento	Observando los distintos caminos a elegir en la aplicación, que la información está organizada.

Tabla2: Requisito funcional RSF_01

ID	RSF_02
Tipo	Funcional
Patrón(es)	D2 (<i>Content modules</i>), B8 (<i>Category pages</i>), K3 (<i>Tab Rows</i>)
Descripción	Se realiza una división del contenido en dos grandes grupos para las distintas secciones de manera que se consiga evitar que la información aparezca mezclada y poco clara
Importancia	Vital
Justificación	Sin este requisito no se podrían establecer módulos para los distintos apartados y la información estaría reflejada sin ningún orden.
Criterio de cumplimiento	Observando los distintos caminos a elegir en la aplicación, que la información está organizada.

Tabla3: Requisito funcional RSF_02

ID	RSF_03
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	Para comenzar a usar la aplicación el usuario deberá seleccionar el tipo de zona a la que se desea acceder, dependiendo de si es administrador o usuario.
Importancia	Importante
Justificación	La aplicación diferencia entre los distintos roles de usuarios según privilegios y controles para mantener la seguridad.
Criterio de cumplimiento	En el formulario principal existen los enlaces para los dos tipos de apartados, no se puede acceder a un apartado concreto sin haber elegido un en el formulario principal.

Tabla4: Requisito funcional RSF_03

ID	RSF_04
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	Para seguir usando la aplicación dentro de la parte del rol de usuario este deberá seleccionar el apartado que desee.
Importancia	Importante
Justificación	La aplicación hace una clara distinción entre los distintos tipos de apartados.
Criterio de cumplimiento	En los formularios consecuentes se deberá seguir el patrón del mismo.

Tabla 5 : Requisito funcional RSF_05

ID	RSF_05
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	El administrador podrá modificar la información desde el gestor de la base de datos
Importancia	Vital
Justificación	La información debe estar controlada y los usuarios no tendrán acceso para no almacenar datos erróneos
Criterio de cumplimiento	La base de datos tiene una contraseña que sólo conoce el administrador.

Tabla 6: Requisito función al RSF_05

ID	RSF_06
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	El usuario podrá estudiar los datos cuando crea necesario.
Importancia	Importante
Justificación	Los datos están alojados en la base de datos, y no recoge nueva información cuando se la ejecuta.
Criterio de cumplimiento	Debe introducir los datos necesarios en los campos pertinentes.

Tabla 7: Requisito Funcional RSF_06

ID	RSF_07
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	Dentro de un tipo de zona el usuario podrá moverse entre las distintas zonas
Importancia	Importante
Justificación	La aplicación realiza una clara distinción entre los dos tipos de zonas
Criterio de cumplimiento	Una vez seleccionada un tipo de zona se puede navegar entre los distintos tipos de manera rápida.

Tabla 8: Requisito funcional RSF_07

ID	RSF_08
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	Dentro de la de zona de administración, para continuar el administrador debe validarse con usuario y contraseña.
Importancia	Importante
Justificación	Las operaciones de esta sección son importantes y para evitar algún error de usuario básico se le protege con cierto control.
Criterio de cumplimiento	EL usuario y la contraseña debe estar alojada en la base de datos principal.

Tabla 9: Requisito funcional RSF_08

ID	RSF_09
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	Dentro de la de zona de administración se deben “Cargar Ficheros” de datos que existen en la aplicación para su estudio.
Importancia	Importante
Justificación	Para ver los ficheros parámetros asociados a estos ficheros y poder “Importar Fichero” que se trata para su estudio.
Criterio de cumplimiento	Dar en el botón “Cargar Fichero”

Tabla 10: Requisito funcional RSF_09

ID	RSF_10
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	Dentro de la de zona de administración se puede “Importar Fichero”. Para poder generar nuevas tabas en la base de datos para nuevos estudios.
Importancia	Importante
Justificación	Para su posterior estudio. También para posibles nuevos ficheros a tratar.
Criterio de cumplimiento	Solo será necesario describir los ficheros parámetros anteriormente en la base de datos.

Tabla 11: Requisito funcional RSF_10

ID	RSF_11
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	Dentro de la de zona de usuario básico se podrán visualizar los” Usuarios Conectados” a la aplicación en la parte de administración.
Importancia	Importante
Justificación	Informarse de quien está conectado a la aplicación.
Criterio de cumplimiento	El campo de conexión se activa cuando un usuario se valida como administrador.

Tabla 12: Requisito funcional RSF_11

ID	RSF_12
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	El usuario puede ver las tablas del sistema en “Ver Tablas”.
Importancia	Importante
Justificación	De este modo el usuario puede estar informado en cada momento de las tablas del sistema
Criterio de cumplimiento	El campo a rellenar tiene una lista desplegable cuyos valores son los únicos aceptados.

Tabla 13: Requisito funcional RSF_12

ID	RSF_13
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	El usuario puede ver los errores producidos en el servidor Apache. Los errores están identificados por el tipo PHP. Está acotado a las horas de mayor actividad entre las 18:00 y 24:00.
Importancia	Importante
Justificación	De este modo el usuario puede ver la información en formato gráfico rápidamente. Se despliega un navegador web con la información en sistema de coordenadas.
Criterio de cumplimiento	El usuario debe pulsar el botón 1 del apartado de control del servidor.

Tabla 14: Requisito funcional RSF_13

ID	RSF_14
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	El usuario puede seleccionar el tipo de error que desee (notice, warn, error). Este estudio será de tres años (2009/2010/2011).
Importancia	Importante
Justificación	De este modo se puede tener la información de forma directa y resumida.
Criterio de cumplimiento	Se debe seleccionar el tipo de error que se quiere estudiar.

Tabla 15: Requisito funcional RSF_14

ID	RSF_15
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	El usuario puede navegar por cada una de las opciones dentro del apartado “control de usuarios”, tanto en Operarios de taller como Usuarios del servidor snoopy. En cada modulo de estos debe rellenar la información que se le pide.
Importancia	Importante
Justificación	De este modo el usuario puede seleccionar la información que necesite de forma clara y rápida.
Criterio de cumplimiento	Los campos a rellenar siguen unas condiciones, la aplicación será la encargada de controlar el posible error del usuario.

Tabla 16: Requisito funcional RSF_15

ID	RSF_16
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	El usuario puede ver las tablas del sistema en “Ver Tablas”.
Importancia	Importante
Justificación	De este modo el usuario puede estar informado en cada momento de las tablas del sistema
Criterio de cumplimiento	El campo a rellenar tiene una lista desplegable cuyos valores son los únicos aceptados.

Tabla 17: Requisito funcional RSF_16

ID	RSF_17
Tipo	Funcional
Patrón(es)	<no aplica>
Descripción	El usuario podrá introducir información para indicar al administrador que quiere hacer un nuevo estudio.
Importancia	Importante
Justificación	De este modo el usuario puede peticionar nuevos estudios. Para futuros nuevos ficheros o nuevas ejecuciones de los mismos actualizados.
Criterio de cumplimiento	Los campos a rellenar deben ser cumplimentados.

Tabla 18: Requisito funcional RSF_17

Requisitos No Funcionales

ID	RSNF_12
Tipo	No Funcional
Descripción	El usuario tiene información contextual en los gráficos.
Importancia	Importante
Justificación	Ayuda a leer el gráfico.
Criterio de cumplimiento	Tener varios en pantalla y que se pueda orientar rápidamente el usuario.

Tabla 19: Requisito no funcional RSNF_12

Requisitos de Usabilidad:

ID	RSU_01
Tipo	Usabilidad
Patrón(es)	K10 (Obvious links), BI1 (Action buttons), AI1 (Interaction)
Descripción	Todos los enlaces utilizados para relacionar los distintos módulos de la aplicación estarán representados de una manera especial para que sean fácilmente distinguidos por los usuarios
Importancia	Vital
Justificación	Es importante no confundir al usuario
Criterio de cumplimiento	Todos los enlaces están debidamente nombrados

Tabla 20: Requisito de Usabilidad RSU_01

ID	RSU_02
Tipo	Usabilidad
Patrón(es)	MI1 (Information on demand)
Descripción	La información se representa de manera que el usuario pueda observarla en su totalidad, sin la necesidad de utilizar las barras de desplazamiento
Importancia	Importante
Justificación	Se muestra al usuario la información de manera clara y concisa
Criterio de cumplimiento	No hay barras de desplazamiento en las páginas principales

Tabla 21: Requisito de usabilidad RSU_02

ID	RSU_03
Tipo	Usabilidad
Patrón(es)	MI1 (Information on demand)
Descripción	La información queda representada de forma que los usuarios puedan unir las principales categorías de información de una manera secuencial y ordenada
Importancia	Vital
Justificación	Se recuerda al usuario los pasos previos hasta alcanzar la situación actual, de manera que favorece la interacción del usuario
Criterio de cumplimiento	Se puede observar la situación del usuario dentro de la jerarquía.

Tabla 22: Requisito de Usabilidad RSU_03

ID	RSU_04
Tipo	Usabilidad
Patrón(es)	MI2 (Process Feedback)
Descripción	En el proceso en el que la aplicación está cargando los distintos mapas que debe mostrar al usuario, se refleja un mensaje para proporcionar información en todo momento de los procesos internos que se están realizando.
Importancia	Importante
Justificación	Tanto en búsquedas, como a la hora de mostrar mucho contenido informativo referente a una base de datos, y en procesos de gestión de archivos, mostraremos mensajes al usuario sobre el estado del proceso en cuestión para que no abandone la página pensando que ha habido un error en la aplicación.
Criterio de cumplimiento	Mensaje informativo.

Tabla 23: Requisito de usabilidad RSU_04

ID	RSU_05
Tipo	Usabilidad
Patrón(es)	MI1 (Information on demand)
Descripción	La información queda representada de forma que los usuarios puedan unir las principales categorías de información de una manera secuencial y ordenada
Importancia	Vital
Justificación	Se recuerda al usuario los pasos previos hasta alcanzar la situación actual, de manera que favorece la interacción del usuario
Criterio de cumplimiento	Se puede observar la situación del usuario dentro de la jerarquía.

Tabla 24: Requisito de usabilidad RSU_05

ID	RSU_06
Tipo	Usabilidad
Patrón(es)	L6 (Fast-Loading Contents)
Descripción	Se minimizará el tiempo máximo posible de la carga de la aplicación
Importancia	Vital
Justificación	Se optimizará el código para que la carga se realice en el menor tiempo posible
Criterio de cumplimiento	Código optimizado en funciones independientes y claramente diferenciables

Tabla 25: Requisito de usabilidad RSU_06

ID	RSU_07
Tipo	Usabilidad
Patrón(es)	AP1 (Aesthetics), B9 (Site accessibility)
Descripción	La aplicación tendrá una gama de colores acordes con los temas que abarca y que faciliten la lectura
Importancia	
Justificación	Se utilizan unos colores que resulten agradables para el usuario y ayuden a la navegación a personas con algún tipo de discapacidad visual
Criterio de cumplimiento	Evitar colores combinar colores fuertes y demasiado luminosos

Tabla 26: Requisito de usabilidad RSU_07

ID	RSU_08
Tipo	Usabilidad
Patrón(es)	B3 (Hierarchical organization)
Descripción	La aplicación tendrá una organización jerárquica
Importancia	Vital
Justificación	Las jerarquías ayudan a tener una visión global del sistema
Criterio de cumplimiento	Observar que en algunas secciones hay jerarquías

Tabla 27: Requisito de usabilidad RSU_08

ID	RSU_09
Tipo	Usabilidad
Patrón(es)	D1 (Page Templates)
Descripción	Se debe usar una misma plantilla para todas las instancias de la aplicación
Importancia	Importante
Justificación	Proporcionarán uniformidad y fácil uso para el usuario.
Criterio de cumplimiento	Se debe estandarizar el diseño de la aplicación. Comprobar que todas las instancias cumplen el mismo patrón.

Tabla 28: Requisito de usabilidad RSU_09

ID	RSU_10
Tipo	Usabilidad
Patrón(es)	A5 (About as)
Descripción	Utilizar nombres descriptivos en los enlaces e hipervínculos.
Importancia	Importante
Justificación	De esta manera los usuarios tendrán una noción de donde van a dirigirse.
Criterio de cumplimiento	Lo mostrado tiene que ser lo descrito en el hipervínculo y son entendibles.

Tabla 29: Requisito de usabilidad

ID	RSU_11
Tipo	Usabilidad
Patrón(es)	K1 (Familiar Language)
Descripción	El lenguaje que se utiliza se comprensible para el usuario al que va destinada la aplicación.
Importancia	Importante
Justificación	Esto ayuda que los usuarios comprendan la aplicación fácilmente.
Criterio de cumplimiento	Comprobar que el lenguaje utilizado es comprensible.

Tabla 30: Requisito de usabilidad RSU_11

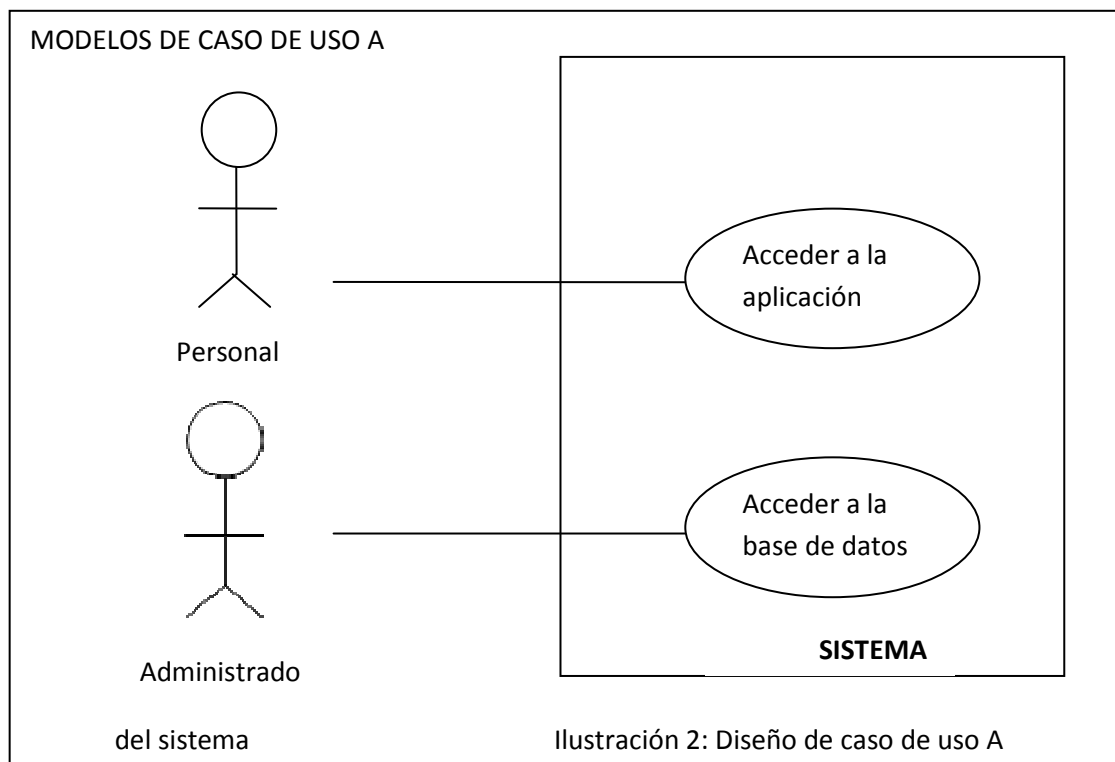
3.1.2 Casos de uso

Un caso de uso es una técnica usada para realizar la captura de requisitos de un sistema informático. Cada caso de uso muestra los escenarios que indican cómo se interactúa entre el sistema y los usuarios del mismo. En este caso tenemos dos tipos de usuarios diferenciados:

- Personal: Aquellas personas que pertenecen a la organización en la que se utilice esta aplicación. Personas expertas que interpretan los datos.
- Administradores del sistema: Aquella con permisos necesarios para llevar a cabo las funciones administrativas y con acceso a la base de datos para sus posibles modificaciones.

3.1.2.1 Descripción gráfica

En este apartado se muestran los diagramas de casos de uso representativos de las principales funcionalidades de los usuarios de la aplicación.



MODELO DE CASOS DE USO A.2

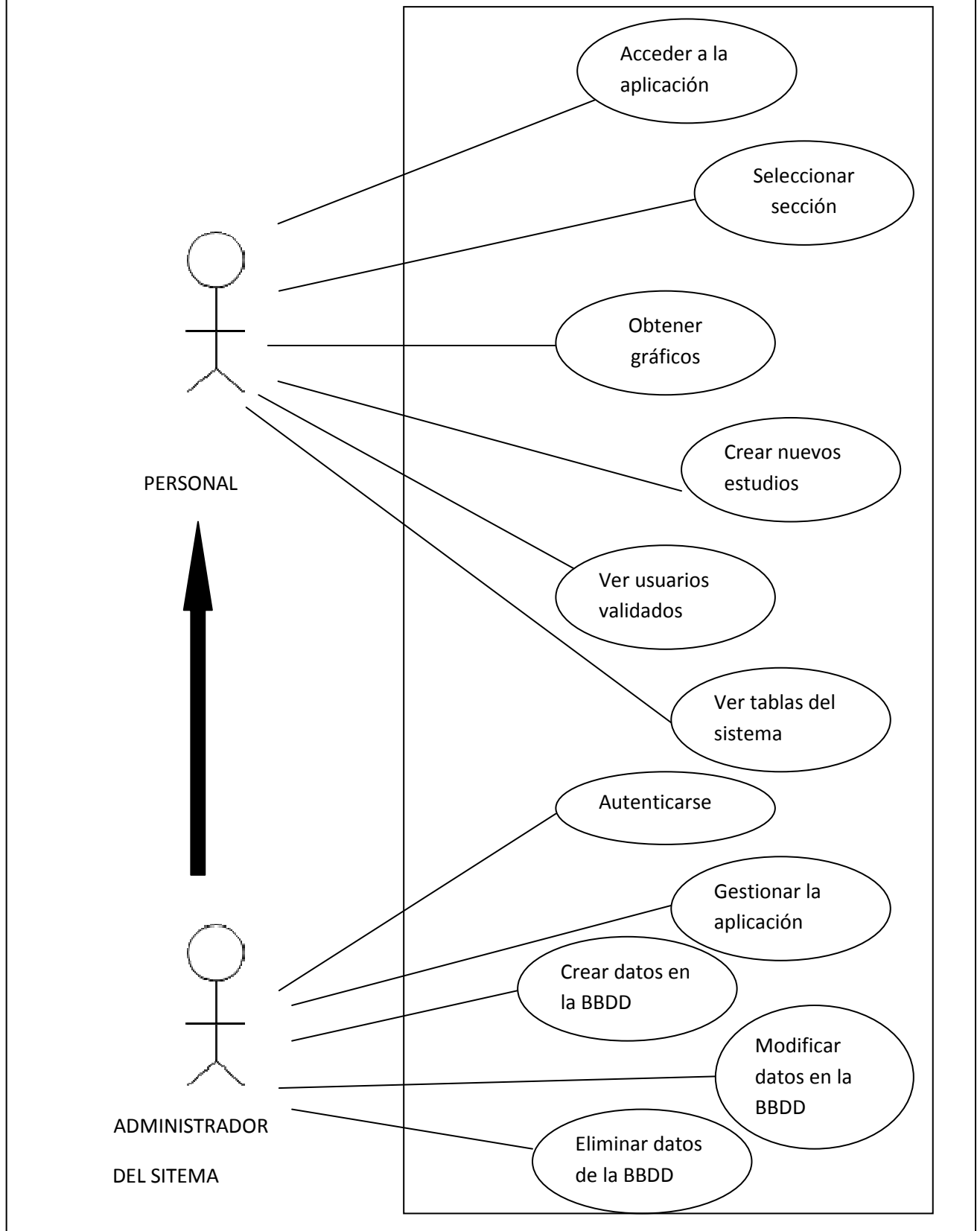


Ilustración 3: Modelo de caso de uso A.2

3.1.2.2 Descripción textual

Para cada uno de los casos de uso identificados en el apartado anterior se realizará una descripción textual para aclarar las funcionalidades. Los campos que describen los casos de uso son los siguientes:

- **Identificador:** Nombre de manera univoca un caso de uso. La nomenclatura a seguir será CU-XX siendo XX números comprendidos entre 0 y 9 que se irán incrementando consecutivamente.

- **Nombre:** Descripción breve del caso de uso.

* Tanto el campo *Identificador* como *Nombre* no estarán identificados en la tabla como campos sino que formarán parte del título de las mismas.

- **Actor:** Agente externo que interacciona con el sistema.

- **Objetivo:** Breve explicación del caso de uso.

- **Precondiciones:** Condiciones que deben cumplirse para poder ejecutar el caso de uso.

- **Pos condiciones:** Condiciones que se producen tras la ejecución del caso de uso.

- **Escenario básico:** Interacción más típica entre actor y sistema detallando la información y los cambios observados en el sistema.

- **Escenario alternativo:** Ejecución del caso de uso con condiciones de error o caminos de decisión distintos al básico

CU.01	Seleccionar el tipo de zona
Actor	Personal
Objetivo	Acceder a su pantalla de la aplicación
Precondiciones	El usuario debe abrir la aplicación.
Pos condiciones	El usuario se encuentra su pantalla de usuario
Escenario básico	<ol style="list-style-type: none"> 1. Se accede a la página principal de la aplicación 2. Selección el botón que referencia a cada tipo de usuario
Escenario alternativo	<ol style="list-style-type: none"> 3. Se selecciona el botón de salir para cerrar la aplicación

Tabla 31: Descripción caso de uso CU.01

CU.02	Seleccionar sección
Actor	Personal
Objetivo	Se accede a la zona con la información concreta de cada ejemplo.
Precondiciones	El personal a selecciona su tipo de zona.
Pos condiciones	Entra en la parte concreta a estudiar, parca cada uno de los ejemplos propuestos y varias opciones concretas.
Escenario básico	<ol style="list-style-type: none"> 1. Selecciona uno de los ejemplos
Escenario alternativo	<ol style="list-style-type: none"> 2. Selección alguno de los botones de información 3. Selección de salir para volver a la pantalla principal

Tabla 32: Descripción caso de uso CU.02

CU.03	Obtener gráficos
Actor	Personal
Objetivo	Obtener los gráficos del estudio
Precondiciones	EL usuario selecciona el botón de la zona que se quiere obtener el grafico.
Pos condiciones	Se obtiene el gráfico de la sección que se está estudiando en la zona seleccionada
Escenario básico	<ol style="list-style-type: none"> 1. Se selecciona uno de los botones 2. Se introducen los datos correctos en cada una de las siguientes secciones.
Escenario alternativo	<ol style="list-style-type: none"> 3. Volver atrás

Tabla 33: Descripción caso de uso CU.03

CU.04	Crear nuevos estudios
Actor	Personal
Objetivo	Crear nuevos estudios
Precondiciones	No acceder a los botones de estudios de las distintas secciones
Pos condiciones	Obtener un fichero parametrizador para preparar un nuevo estudio con nuevos ficheros de datos
Escenario básico	<ol style="list-style-type: none"> 1. Selección del botón “Configurar” 2. Rellenar los campos de datos
Escenario alternativo	<ol style="list-style-type: none"> 3. Volver atrás

Tabla 34: Descripción caso de uso CU.04

CU.05	Ver usuarios validados
Actor	Personal
Objetivo	Poder visualizar los usuarios validados en la parte de administrador.
Precondiciones	Estar en la pantalla principal del usuario, pinchar en el botón de la pantalla “Usuarios Conectados”
Pos condiciones	Obtener una pantalla con la información solicitada.
Escenario básico	Pinchar en el botón “Usuarios Conectados”
Escenario alternativo	Pinchar en el botón volver de la ventana que aparece.

Tabla 35: Descripción caso de uso CU.05

CU.06	Ver tablas del sistema
Actor	Personal
Objetivo	Obtener una pantalla con la lista de las tablas del sistema.
Precondiciones	Estar en la pantalla principal de usuario. Pinchar en el botón “Ver tablas”
Pos condiciones	Se obtiene una pantalla con una lista incluyendo dentro de ella las nomenclaturas de las tablas existentes.
Escenario básico	1. Entrar en la pantalla del usuario 2. Pinchar en el botón “Ver tablas” 3. Pinchar en “Cargar Tablas”
Escenario alternativo	4. Pinchar en el botón “Volver”

Tabla 36: Descripción caso de uso CU.06

CU.07	Acceder a la aplicación
Actor	Administrador del sistema
Objetivo	Acceder a la parte administrativa del sistema
Precondiciones	Estar en el Inicio de la aplicación y validarse como usuario administrador del sistema. Con usuario y contraseña.
Pos condiciones	Obtener la parte administrativa en el formulario
Escenario básico	<ol style="list-style-type: none"> 1. Se accede al apartado del servidor (localhost) 2. Se introduce un nombre de usuario y contraseña válidas 3. Si la autenticación es correcta se accede a la parte privada 4. Se muestran los campos de administración
Escenario alternativo	<ol style="list-style-type: none"> 5. Salir sin hacer nada

Tabla 37: Descripción caso de uso CU.07

CU.08	Gestionar la aplicación
Actor	Administrador del sistema
Objetivo	Gestionar los datos del sistema
Precondiciones	Validarse en la aplicación.
Pos condiciones	Obtener las tablas del sistema Cargando las tablas
Escenario básico	<ol style="list-style-type: none"> 1. Se accede al apartado del servidor (localhost) 2. Se introduce un nombre de usuario y contraseña válidas 3. Si la autenticación es correcta se accede a la parte privada 4. Pulsar el botón “Cargar Tablas”
Escenario alternativo	<ol style="list-style-type: none"> 5. Pulsar el botón salir para no hacer nada

Tabla 38: Descripción caso de uso CU.08

CU.09	Crear datos en la BBDD
Actor	Administrador del sistema
Objetivo	Gestionar los datos del sistema, cargando de nuevo los ficheros de datos.
Precondiciones	Validarse en la aplicación.
Pos condiciones	Importar los ficheros para cargar los nuevos datos en las tablas de la base de datos
Escenario básico	<ol style="list-style-type: none"> 1. Se accede al apartado del servidor (localhost) 2. Se introduce un nombre de usuario y contraseña válidas 3. Si la autenticación es correcta se accede a la parte privada 4. Pulsar el botón “Cargar Tablas” y después “Importar Ficheros”
Escenario alternativo	<ol style="list-style-type: none"> 5. Pulsar el botón salir para no hacer nada

Tabla 39: Descripción caso de uso CU.09

CU.10	Modificar datos en la BBDD
Actor	Administrador del sistema
Objetivo	Gestionar los datos del sistema, cargando de nuevo los ficheros de datos. No se crean nuevos datos es modificarlos
Precondiciones	Validarse en la aplicación.
Pos condiciones	Importar los ficheros para cargar los nuevos datos en las tablas de la base de datos
Escenario básico	<ol style="list-style-type: none"> 1. Se accede al apartado del servidor (localhost) 2. Se introduce un nombre de usuario y contraseña válidas 3. Si la autenticación es correcta se accede a la parte privada 4. Pulsar el botón “Cargar Tablas” y después “Importar Ficheros”

Escenario alternativo	5. Pulsar el botón salir para no hacer nada
------------------------------	---

Tabla 40: Descripción caso de uso CU.10

CU.11	Eliminar datos en la BBDD
Actor	Administrador del sistema
Objetivo	Gestionar los datos del sistema, cargando de nuevo los ficheros de datos.
Precondiciones	Validarse en la aplicación. Los ficheros a cargar deben estar vacíos.
Pos condiciones	Importar los ficheros para cargar los nuevos datos en las tablas de la base de datos
Escenario básico	<ol style="list-style-type: none"> 1. Se accede al apartado del servidor (localhost) 2. Se introduce un nombre de usuario y contraseña válidas 3. Si la autenticación es correcta se accede a la parte privada 4. Pulsar el botón “Cargar Tablas” y después “Importar Ficheros”
Escenario alternativo	5.Pulsar el botón salir para no hacer nada

Tabla 41: Descripción de caso de uso CU.11

3.2 Diseño

3.2.1 Base de datos

3.2.1.1 Modelo E/R

El modelo E/R o Entidad-Relación es una herramienta utilizada para el modelado de los datos en un sistema de información. En el diagrama se reflejan las entidades relevantes del sistema así como sus interrelaciones y propiedades. El objetivo del modelo entidad-relación es representar los objetos de la base de datos como entidades con atributos y que se vinculan entre ellas mediante relaciones.

Se considera entidad a cualquier tipo de objeto o concepto sobre el que se recoge información. Las entidades se representan gráficamente mediante rectángulos y su nombre, identificativo y unívoco, aparece en el interior.

Las entidades están formadas por atributos. Un atributo es una característica de interés o un hecho sobre una entidad o sobre una relación.

Los atributos representan las propiedades básicas de las entidades y de las relaciones.

Gráficamente se representan mediante bolitas que cuelgan de las entidades o relaciones a las que pertenecen. Como breve explicación de los atributos representados en el gráfico podemos decir que los atributos principales se representan con círculo negro y los opcionales con una línea discontinua. Otro tipo de atributos son los multievaluados, es decir, se pueden almacenar diferentes datos en un mismo atributo, estos serán representados con la línea continua y una flecha.

Una relación es la correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre identificativo aparece en el interior. Un tipo especial de relación que se ha utilizado en el modelo es la herencia, la herencia es un tipo de relación entre entidad "padre" y una entidad "hijo". La entidad "hijo" hereda todos los atributos y relaciones de la entidad "padre". Esta relación se representa mediante un triángulo, la entidad "padre" se encuentra en el vértice superior del triángulo y las entidades "hijo" se conectarán por la base del triángulo.

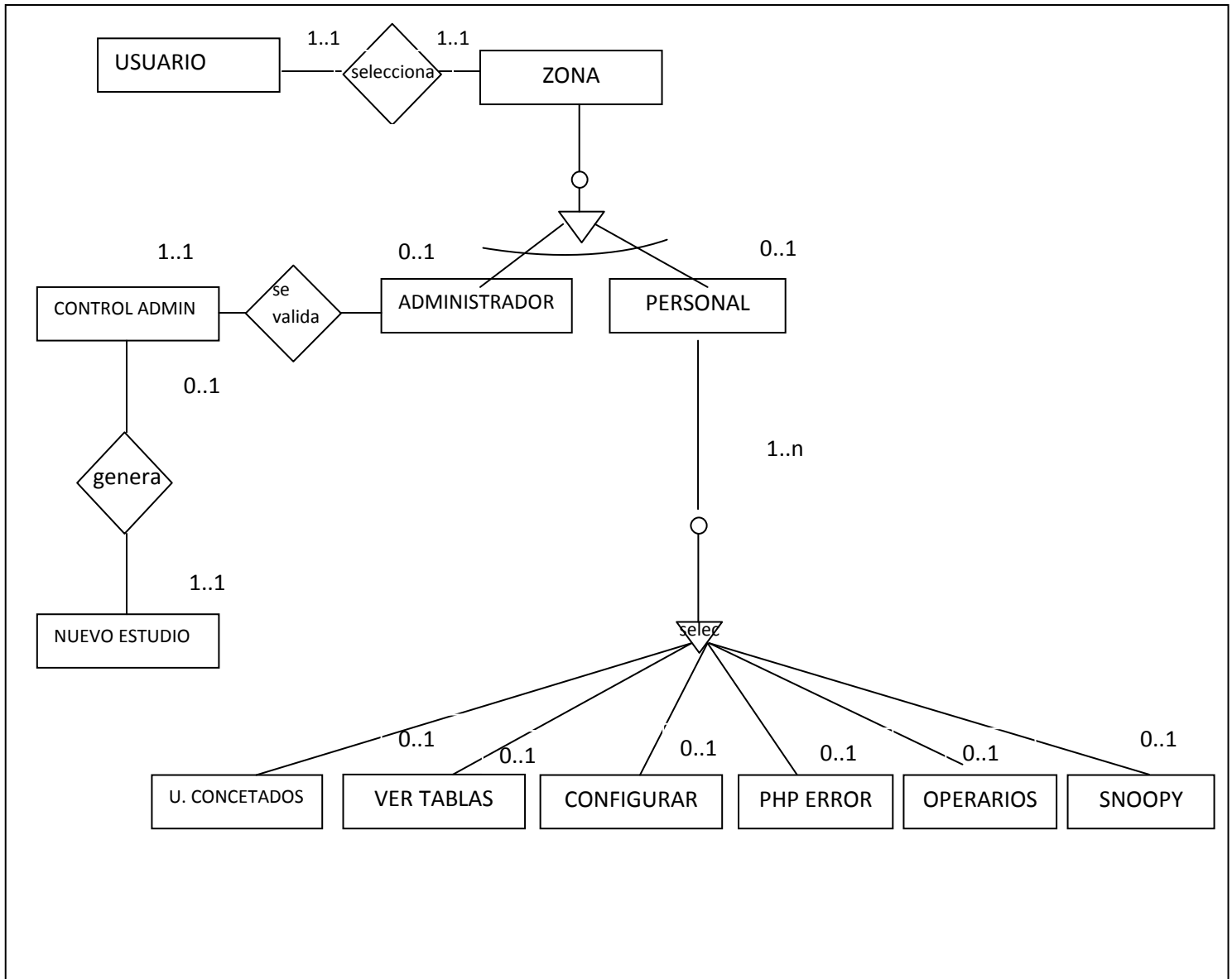


Ilustración 4: Modelo E/R

Los distintos usuarios seleccionan una zona concreta de la aplicación bien sean administrador o personal de la empresa. Si un usuario del tipo personal selecciona el de tipo administrador y no está validado no accederá a esa parte del sistema por no tener los privilegios suficientes. Al contrario el administrador validado podrá acceder y generar nuevos estudios.

El personal de la empresa accede a las distintas secciones de la empresa. Estas secciones están subdivididas en diez apartados, Ver Tablas, en la que podrá observar las distintas tablas de la base de datos que existen en el sistema. Ver los usuarios conectados y validados administradores del sistema. Configurar un nuevo fichero parametrizador, observar los distintos estudios realizados en los que se engloba los seis tipos de gráficos que se pueden obtener (php_error apache, php_operarios y php_snoopy, cada uno con dos variantes)

Para configurar un nuevo fichero parametrizador, el personal debe introducir los nombres de cada uno de los componentes que forman el fichero. Esto será utilizado por el administrador para generar nuevos estudios en la parte de administración.

Para observar uno de los estudios ya creados en la aplicación, el personal debe seleccionar aquel que le interese en cada uno de los botones destinados para ello. Alguno de ellos solicita el ingreso de datos relacionados con el estudio. Estos datos pueden variar en horas iniciales y finales o nombres de operarios por ejemplo.

Se pueden crear varios gráficos de un mismo estudio cambiando los parámetros solicitados anteriormente o bien sólo uno como en los gráficos de php error que ya están definidos por el sistema.

3.2.1.1 Modelo Relacional

El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. Su idea fundamental es el uso de las relaciones.

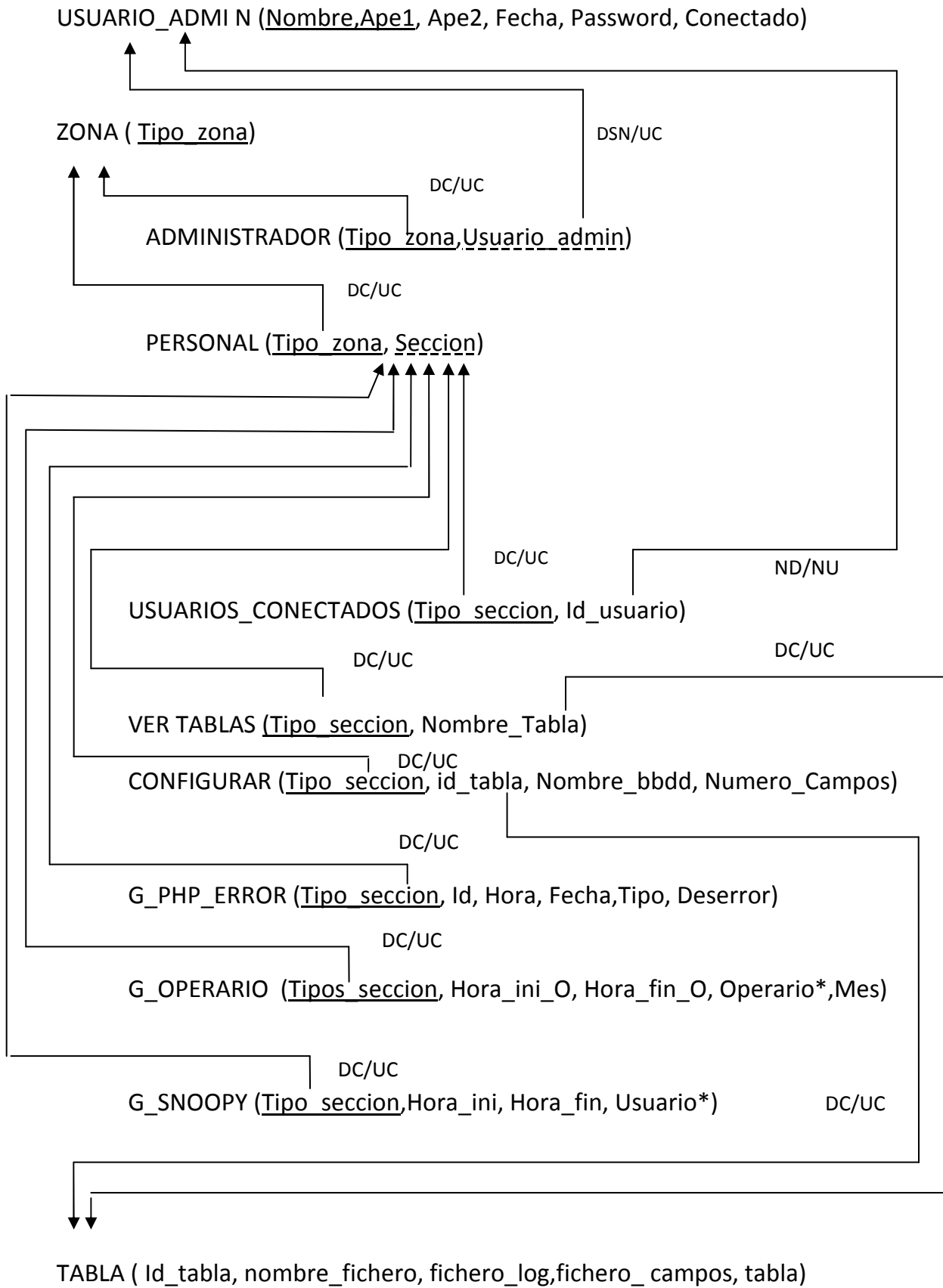
El modelo relacional fue propuesto por E.F. Codd en los laboratorios de IBM en California. Se trata de un modelo lógico [Irene Luque Ruiz - Ed. Ra-ma] que establece una estructura sobre los datos, aunque éstos posteriormente puedan ser almacenados de múltiples formas para aprovechar características físicas concretas de la máquina sobre la que se implante la base de datos realmente.

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en que estos se almacenen no tiene relevancia. Esto tiene como ventaja la facilidad para entender y utilizar por un usuario no experto, la información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

A continuación se van a definir las pautas seguidas para la representación del esquema que ayudarán al entendimiento del mismo:

- las relaciones son representadas en **negrita**, con un identificador único y entre paréntesis se sitúan los atributos que contendrá cada dupla de esa relación.
- Los atributos de cada relación están separados por comas, pueden ser opcionales y no tomar ningún valor, en tal caso se indicará con un asterisco.
- Aquellos atributos que sean clave primaria de la tabla aparecerán subrayados, y los que representen clave ajena aparecerán con una flecha inferior que estará apuntando a la tabla a la que ese atributo está diferenciando. A cada flecha le estará asignado un texto que definirá las reglas de borrado y modificación de las claves ajenas. Por un lado la regla de borrado elegida es (*DC-delete cascade*) que sin duda es la menos restrictiva, excepto en uno de los casos en el que hemos elegido (*DSN-delete set null*) donde la tabla a borrar no será eliminada, los valores no válidos serán sustituidos por valores nulos.

Por otro lado, la regla de modificación se realizará en todos los casos en cascada (*UC-update cascade*).



USUARIO_ADMIN: Esta tabla recoge los datos de los usuarios de tipo Administrador. Son los únicos usuarios que tienen datos almacenados en el sistema, los demás usuarios entran libremente a la aplicación sin tener ningún registro sobre ellos.

El usuario está identificado por un nombre y un apellido, un usuario puede tener el mismo nombre o apellido pero nunca los dos juntos. Por eso, aparecen subrayados. Además son el índice de la tabla. El usuario administrador debe tener asociado una contraseña y un marcador de estado conectado.

- Nombre: Indica el nombre del usuario administrador. Es de tipo varchar, este tipo engloba texto alfabético. Es parte de la clave principal de la tabla. Es clave principal por lo que es único junto al apellido uno.
- Ape1: Identifica al administrador unido al campo anterior. Es también de treinta caracteres, se le supone suficiente para identificar al campo. Si existiera un apellido más largo se truncaría. Tampoco puede ser nulo.
- Ape2: Identifica el segundo apellido del usuario administrador. Es de tipo varchar como los anteriores. Puede ser nulo en este caso
- Fecha: Identifica la fecha de nacimiento del individuo. Esta fecha tiene formato número para tratarla mejor en el desarrollo del programa. Puede estar vacía.
- Password: Identifica la clave personal del usuario administrador. Sin esta clave no se puede acceder a la siguiente sección, por lo que no puede ser nula.
- Conectado: Describe el estado en el sistema del usuario administrador. Es de tipo *integer*, el valor cero significa desconectado y el valor uno conectado. Este valor no puede ser nulo o es uno o el otro.

ZONA: Tabla que identifica la zona en la que se va acceder. Si se escoge la zona administrador se dirigirá hacia la sección o en el caso contrario a la otra. Se identifica esto con el campo tipo de zona que compone la tabla.

Tipo Zona: Identifica la zona de usuario, es la clave primaria de la tabla; Administrador o Personal. Es de tipo entero de dos dígitos máximo. No puede tomar valores nulos

ADMINISTRADOR: Es la tabla “hijo” de la tabla Zona, por tanto tendrá atributos propios y atributos de la tabla padre, como es Tipo Zona. Contiene una clave primaria heredada de zona y un clave ajena que apunta hacia Usuario Administrador.

Tipo Zona: Heredada de Zona e identificativa conjunta con el campo Usuario admin. Tipo Varchar y no puede ser nula.

Usuario_admin: Calve ajena y primaria, apuntada hacia la tabla usuario administrador. Todo usuario administrador debe tener unos datos propios. Es del tipo varchar de 30, tanto el nombre y el apellido como se muestra en la tabla usuario admin. No puede ser nula.

PERSONAL: Es la tabla “hijo” de la tabla Zona. Contiene el campo identificador tipo de zona y el campo propio y clave ajena Sección. Esta marcará a sección a la que se dirige el usuario tipo personal.

Tipo zona: Clave identificativa al igual que sucede con el mismo campo de administrador.

Sección: Clave ajena para identificar la sección. Es el campo que identifica a la sección que el usuario personal accede.

USUARIOS CONECTADOS: Tabla “hija” que identifica los usuarios conectados y validados en la aplicación. Estos usuarios son de tipo administrador. Esto lo consigue con la clave ajena `id_usuario`. Esta tabla es para informar al usuario personal que administradores están en el sistema para solicitarles alguna sugerencia o pedirles un nuevo estudio.

Tipo_sección: Es la clave primaria de la tabla. De tipo `varchar` de tres dígitos. No puede ser nula.

Id_usuario: Clave ajena que apunta a la tabla administradores. Contiene la información de los usuarios administradores del sistema. Al ser una clave referenciada contiene los datos de la tabla a la que apunta.

VER TABLAS: Tabla “hija” de la tabla sección. Identifica las tablas que existen en el sistema. Contiene un campo identificador y otro que identifica las tablas.

Tipo_seccion: Es la clave primaria de la tabla. No puede ser nula y es de tipo `varchar` de 2 caracteres. Identifica la sección a la que accede el usuario.

Nombre Tabla: Identificador que contiene el nombre de las tablas. Es una clave ajena que toma el campo nombre de la tabla Tabla.

CONFIGURAR: Esta tabla almacena la información introducida por el usuario personal para un nuevo estudio. Contiene un campo identificador Tipo, “calve hija” de la tabla Sección. Además contiene el la clave ajena `id_tabla`, este campo apunta a la tabla Tabla, para tener toda la información necesaria. También aloja el nombre de la base de datos y el número de campos para el nuevo estudio.

Tipo_seccion: Calve primaria y ajena, es la calve hija derivada de la tabla sección. Es de tipo `varchar` de dos dígitos como en los anteriores casos

Id_tabla: Clave ajena identificadora de la tabla en el nuevo estudio. De tipo *integer* de veinte dígitos. No puede estar vacío.

NombreBaseDatos: Campo identificador del nombre de la base de datos. En este caso, es una constante, puesto que sólo existe una base de datos.

Numero de campos: Numero de campos que formarán la nueva tabla del estudio. No puede estar vacío. De tipo *integer* de dos dígitos.

G_PHP_ERROR: identifica la tabla tb_php_error de la base de datos y servirá para representar el gráfico php error. Está identificado por un campo Tipo_g que describe el tipo de sección de la que proviene, un campo identificador clave primaria , una fecha, hora, tipo de error y la descripción del error.

Tipo_seccion: Campo que identifica al grafico. Es de tipo *integer* de tres dígitos como máximo. No puede ser nulo porque es la clave identificativa.

ID: Clave primaria identificativa de la tabla. Es de tipo entero de cómo máximo cincuenta dígitos. No puede ser nula. Este campo se autoincrementa el sólo automáticamente. Identifica cada uno de los registro de la tabla.

Fecha: Identificador de la fecha en la que se creó el registro. Es de tipo varchar de 10 dígitos. No puede ser nulo.

Hora: Identifica la hora en que se produjo el registro. Es de tipo varchar de 8 dígitos. No puede ser nulo.

Tipo: Identifica el tipo de error que se produce. Es de tipo varchar con una longitud máxima de ocho caracteres. No puede ser nulo.

Deserror: Describe el error que se ha producido. Es de tipo varchar y tiene un longitud máxima de mil veinte-cuatro caracteres.

G_OPERARIO: Identifica a la tabla “tb_fichajes” de la base de datos y representa el grafico de Operarios. Almacena la información de los fichajes de los operarios de un taller. Está compuesto de una clave identificativa y unos campos de consulta para obtener la grafica, como la fecha, la hora y el nombre del operario.

Logindex: Es un campo de tipo Único, sirve para ordenar la tabla. Se autoincrementa y es de formato *integer* de once dígitos. No puede ser nulo.

Logtime: Clave primaria e identificativa de la tabla tb_fichajes. Es de tipo *timestamp* y contiene la fecha y la hora del fichaje del operario.

Fecha: Campo de tipo varchar de diez dígitos. Almacena la fecha de fichaje del operario. Es la primera parte del campo logtime. No puede ser nulo.

Hora: Campo de tipo varchar de diez dígitos. Almacena la hora de fichaje del operario. Es la segunda parte del campo logtime. No puede ser nulo.

User_id: Campo de tipo varchar de dieciséis dígitos. Identifica al operario de taller con un número. No puede ser nulo.

G_SNOOPY: Identifica la tabla “tb_php_snoopy”. Esta tabla contiene los datos del comportamiento de un tipo de servidor denominado snoopy. Recoge el día, la hora y la situación del servidor en ese instante.

Ind: Clave primaria de la tabla, de tipo int de veinte dígitos. Se autoincrementa y no puede ser nulo por su condición de única.

Mes: Identifica el mes en que se ha producido el registro. Se representa con tres caracteres del tipo varchar. No puede ser nulo.

Día: Identifica el día en que se produce el registro. Su longitud máxima es de once dígitos, es de tipo integer. No puede ser nulo.

Hora: Representa la hora en la que se produjo el registro. Es de tipo varchar y de longitud máxima de diez caracteres. No puede ser nulo.

Skx: Es un campo con un valor fijo “skx”, de parametrización interna. Es de tipo varchar de diez caracteres. No puede ser nulo.

Comentario: Identifica al usuario del servidor. Es de tipo varchar de veinte caracteres y n puede ser nula.

TABLA: Contiene la información de los ficheros. Está compuesto de los ficheros que se utilizan como parámetros en cada uno de los gráficos. Cada registro identifica el fichero a leer, el fichero campos, fichero constructor y el nombre de la tabla. Contiene también en campo identificativo de cada tabla.

Id_Tabla : Clave identificativa de la tabla. Es clave primaria por lo que no puede ser nula. Es de 20 dígitos de tipo entero. No puede ser nulo.

nombre fichero: Es un campo de texto, con una longitud de cincuenta caracteres. Describe el nombre del fichero constructor que lee el fichero log.

fichero_log: Campo alfanumérico con longitud de cincuenta caracteres. Es el nombre del fichero log a leer por el fichero constructor.

fichero_campos Campo también alfanumérico de cincuenta caracteres. Describe el nombre del fichero que ayuda al fichero constructor a leer el fichero log.

Tabla: Campo de tipo alfanumérico de cincuenta caracteres. Describe el nombre de la tabla que se utiliza en la lectura del fichero log. Es el nombre de la tabla que alojara los datos una vez tratados. Ninguno de los campos puede ser nulo.

3.2.2 Aplicación

3.2.2.1 Diagrama de navegación

Diseñar el diagrama de navegación consiste en describir dinámicamente el orden y tipo de pantallas que se van produciendo durante la ejecución de la aplicación. Consiste en un autómata finito que describe la transición de ventanas para conocer de este modo las ventanas que preceden y a las que se pueden acceder desde cada una de las mismas.

En la siguiente figura se puede observar el diagrama de navegación de la aplicación realizada, de este modo se puede conocer con facilidad el número de ventanas de las que consta nuestra aplicación así como los pasos necesarios para conseguir el fin deseado.

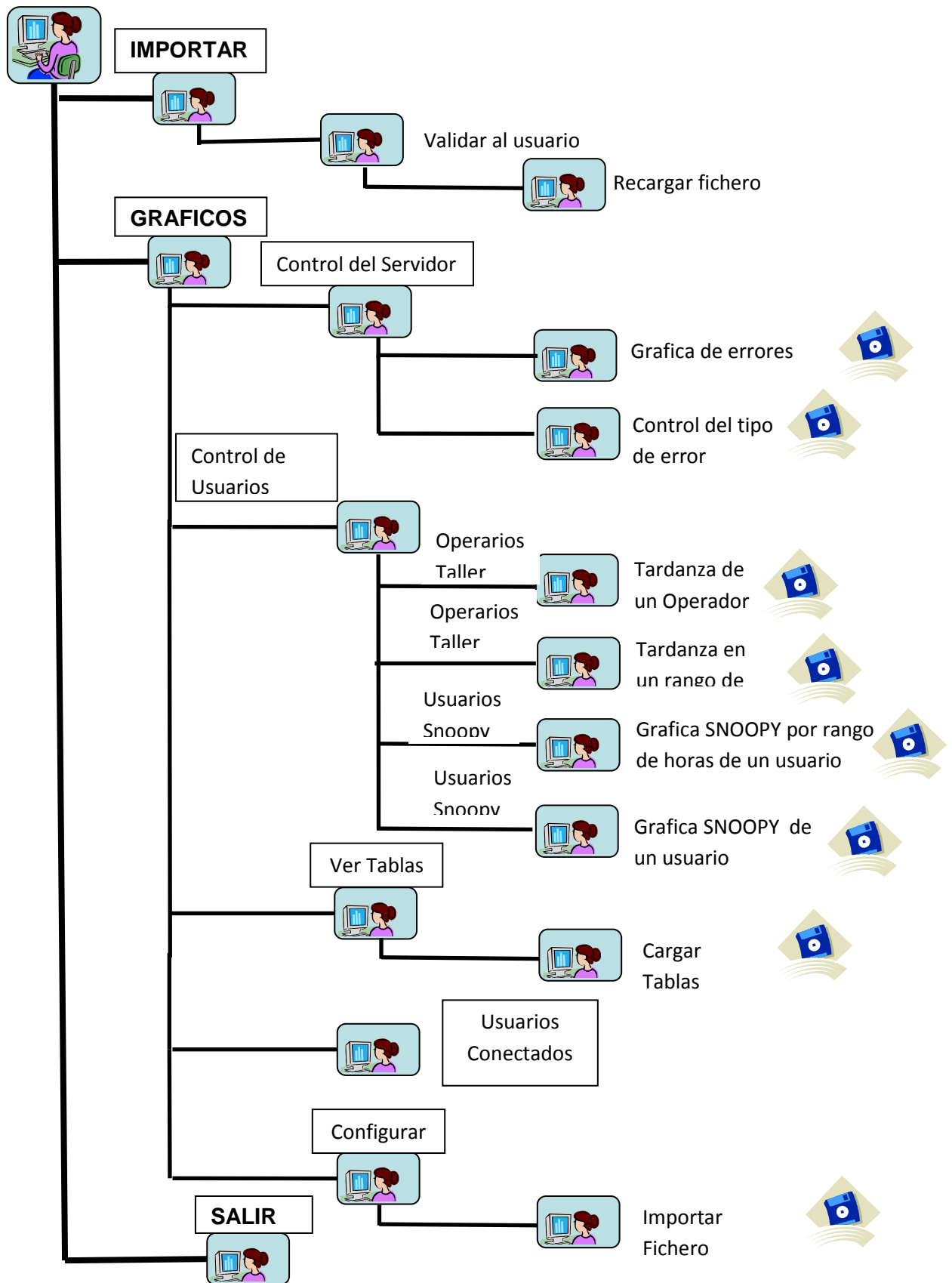


Ilustración 5. Diagrama de navegación

3.3 Implementación

3.3.1 Base de datos

Para tratar y explicar la implementación de la base de datos propuesta hay que referirse al lenguaje SQL. Este ha sido el nexo entre el diseño lógico realizado y el sistema gestor de bases de datos.

Para definir la estructura de la base de datos se ha utilizado el lenguaje LDD y para el manejo de la estructura el LMD. En este punto nos vamos a centrar en los pasos seguidos para la implantación y las principales sentencias utilizadas para especificar la estructura de los datos, por tanto nos centraremos en sentencias del lenguaje LDD.

3.3.1.1 Crear objetos

Partiendo del diseño lógico, el primer paso es crear las tablas en concordancia. Para ello se ha utilizado la sentencia "CREATE TABLE".

```
CREATE TABLE 'TABLA_NOMBRE' (  
    'CAMPO_1' tipo,  
    'CAMPO_2' tipo )
```

Esta sentencia requiere principalmente de tres atributos; un nombre, una definición y unas opciones. Hay numerosas opciones de tabla pero en nuestro caso únicamente la sentencia "Type=InnoDB" para modificar el motor de almacenamiento que usará la tabla y poder así, utilizar claves ajenas.

La definición de la estructura de la tabla se compone de dos partes bien diferenciadas; la declaración de las columnas y las restricciones de claves o índices.

La declaración de columnas consta de tres apartados: el nombre de la columna, el tipo y la posibilidad de indicar si son posibles los valores nulos.

Todas las tablas creadas y los atributos de cada una de ellas han sido definidos y explicados en apartados anteriores. En este punto nos vamos a centrar en definir todos los tipos de datos que han sido necesarios para la implementación:

- *Decimal*: Número en coma flotante, el número se almacena como una cadena y es posible especificar el número de decimales que guardar.

- *SmallInt*: Número entero con o sin signo. Con signo el rango de valores va desde -32768 a 32767. Sin signo, el rango de valores es de 0 a 65535. Su tamaño de almacenamiento es de 2bytes.

- *Integer*: Número entero con o sin signo. Con signo el rango de valores va desde -2147483648 a 2147483647. Sin signo el rango va desde 0 a 429.4967.295. Su tamaño de almacenamiento es de 4bytes.

- *MediumInt*: Número entero con o sin signo. Con signo el rango de valores va desde -8.388.608 a 8.388.607. Sin signo el rango va desde 0 a 16777215. Su tamaño de almacenamiento es de 3bytes.

- *Varchar*: Almacena una cadena de longitud variable. La cadena podrá contener desde 0 a 255 caracteres. Su tamaño de almacenamiento es de $n + 1$ bytes.

- *Enum*: Campo que puede tener un único valor de una lista que se especifica. El tipo Enumerado acepta hasta 65535 valores distintos. Su tamaño de almacenamiento puede ser de 1 o 2 bytes dependiendo del número de valores.

- *Text*: Un texto con un máximo de 65535 caracteres. Su tamaño de almacenamiento es de la longitud más 2 bytes.

- *Timestamp*: Es la combinación de fecha y hora. El rango va desde el 1 de enero de 1970 al año 2037. El formato de almacenamiento depende del tamaño del campo. En nuestro caso, al tener el formato AñoMesDía aammdd el tamaño es de 6 bytes.

- *Date*: Tipo fecha, almacena una fecha. El rango de valores va desde el 1 de enero del 1001 al 31 de diciembre de 9999. El formato de almacenamiento es de año-mes-día y su tamaño de 3 bytes.

En el anexo se puede consultar la sintaxis utilizada para la creación de las tablas en la base de datos que nos ocupa.

3.3.1.2 Consultas

Las consultas son necesarias para obtener la salida de datos en la aplicación Web. Las consultas implementadas para obtener la información de la base de datos son bastante simples y no utilizan ningún operador de conjuntos, debido a que el diseño ha sido realizado para facilitar la implementación.

Para realizar las consultas se ha utilizado el LMD del lenguaje SQL, concretamente el operador 'SELECT', en el cual, indicando el nombre de la columna o columnas que se quieren seleccionar y la tabla de origen de los datos nos permite obtener lo deseado de una manera sencilla.

```
SELECT [DISTINCT] nombre_columna/s  
      FROM nombre_tabla  
      [WHERE {condicion}]  
      [ORDER BY nombre_columna [DESC]]
```

Esta sentencia permite una serie de opciones añadidas a lo anterior. Se pueden indicar premisas para que únicamente se seleccionen registros que cumplan una determinada condición, así como ordenar el resultado de salida, ya sea ascendentemente o descendentemente (añadiendo la palabra 'DESC').

Las consultas del proyecto son muy importantes. No sólo almacenan datos en las tablas de la base de datos o los modifica también es el pilar de su protagonista proyecto. Son fundamentales para que se muestren los distintos gráficos de la aplicación en el servidor local.

Ya sea en código PHP o en Visual Basic 6 son las consultas son formuladas en varias y distintas partes del programa. Se puede observar en los anexos de este manual. Se convierte así en la parte fundamental del proyecto.

3.3.2 Aplicación

En esta sección se explicara paso a paso las partes del proyecto. Se distinguen tres en la explicación. Por una parte se explicará cómo se ha realizado el diseño de la aplicación ejecutable y por las otras acciones relacionadas con el código PHP y la representación gráfica de las consultas.

3.3.2.1 Diseño Visual Basic

La aplicación está diseñada e implementada en visual basic . Se eligió por su competencia para de realizar ventanas-formularios que se conecten con facilidad y se entiendan con tan buenos resultados. Más adelante se podrá comprar porque.

También se convierte en una buena herramienta para incluir distintos objetos que generen o llamen a nuevas aplicaciones, dotándole de un atractivo y competente desarrollo a todo el conjunto.

En la primera pantalla (a partir de ahora llamaremos así a los distintos formularios) el usuario seleccionará la zona. Cada zona se corresponde con un botón, Personal o Administrador. En último caso Salir.

Una vez seleccionado la zona, si el usuario entra en Administración, deberá “loggearse” con su usuario y contraseña asignado por el súper Administrador para continuar. Si es un éxito accederá a la pantalla de administración del sistema. En esta pantalla podrá cargar de nuevos los datos en la base datos. Este proceso consta de; lectura de los ficheros logs proporcionados, tratamiento de los mismo y almacenamiento en la base de datos de ellos. Se debe notar que si el fichero logs es dinámico cada vez que ejecutemos su lectura se cargarán nuevos datos. En el caso de ser estáticos se cargarán siempre los mismos. Esto sucede así porque se querían acotar los datos para no saturar al sistema. El presupuesto para este proyecto es limitado, limitando así su capacidad de almacenamiento.

Los ficheros a leer deben estar con un formato adecuado. El proyecto obtiene a la información de ellos pero no los “limpia”. Hacer una aplicación que limpie los ficheros y después los parametrize, salen del campo de nuestro estudio. Más adelante se explicará en la parte de código php como se realiza el trabajo de importación de los datos.

El usuario Administrador podrá cargar los datos de nuevo en la aplicación seleccionando cada una de las tablas. Estas tablas deben ser cargadas con anterioridad pulsando en el botón “Cargar Tablas”.

Una vez aparezcan en el sistema cada una tiene asociado una serie de parámetros. Estos son el nombre del constructor que se encarga de leer, separar y almacenar los datos. El fichero log , el cual, contiene los datos a leer. El fichero campos que identifica la separación de los campos de una línea del fichero log y los identifica. La base de datos utilizada y por último el nombre de la tabla que se habrá creado en el sistema para alojar los datos leídos.

Con todos estos datos el sistema ejecutará el fichero constructor el cual hará su trabajo realizando una nueva lectura de los datos. En caso se podrá salir sin realizar ningún cambio.

En la parte de personal, a la que se hace sin “loggearse” puesto que es zona publica, se pueden elegir distintas tareas. Se pueden distinguir dos secciones, una para configurar o informar al usuario y la otra para obtener información relacionada con distintos eventos.

La parte izquierda de la pantalla muestra una serie de botones que informan de las tablas del sistema, los usuarios administradores del mismo y un posible estudio que sugerir a los desarrolladores del proyecto.

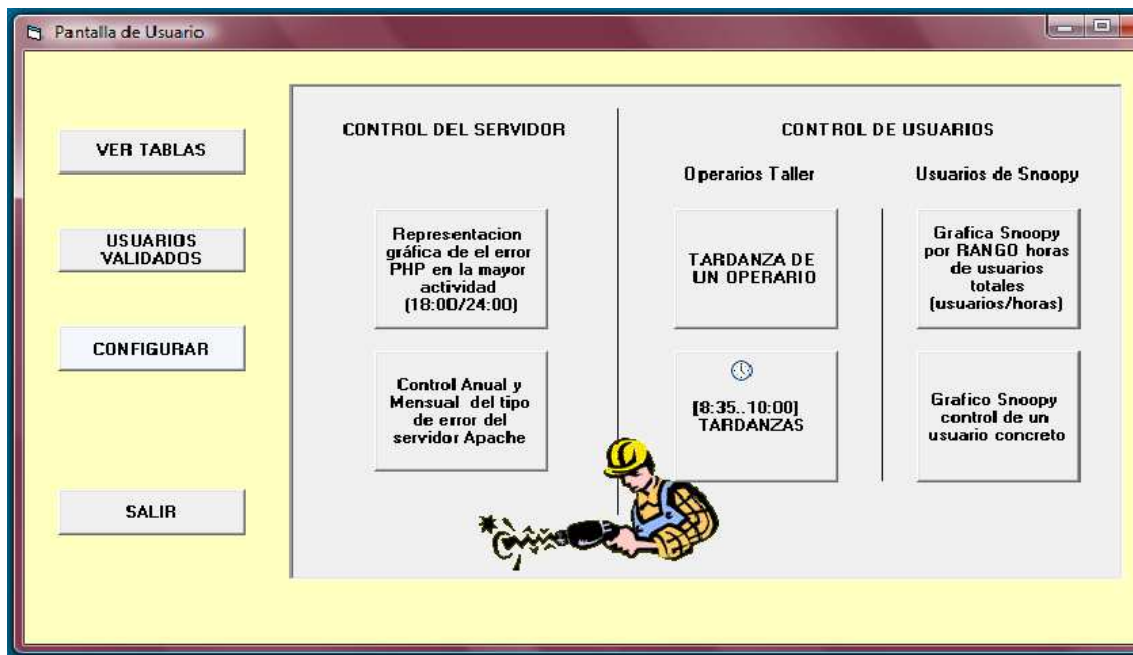


Ilustración 5: Pantalla inicial de usuario

En la parte derecha se muestran los botones que generan los grafico. La rutina principal se muestra más abajo:

```
Set intobj5 = CreateObject("InternetExplorer.Application")

cad = "http://localhost/grafica_apacheerror_juntas.php?tipo=" & tipo

intobj5.Visible = True

intobj5.navigate cad
```

Ilustración 6: Rutina principal Del programa

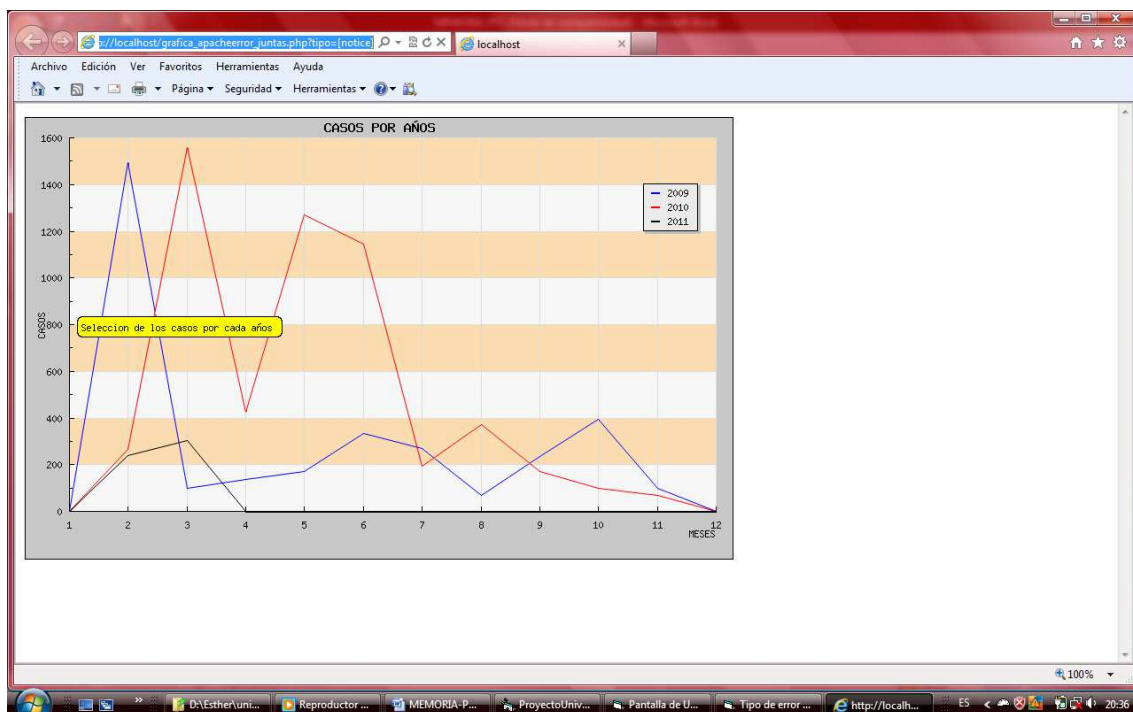


Ilustración 7 : Gráfica resultado

En esta rutina se declara un objeto que llamará a la aplicación de internet explorer. Servirá para m mostrar en pantalla los datos de una forma elegante y rápidamente entendible por el usuario.

Esta página se cargará en servidor web que exista, en el proyecto se ejecuta bajo el localhost, es decir en la misma máquina. El fichero php solicitado se ejecutará y mostrará el grafico en pantalla.

Para conectar con la base de datos se utiliza la rutina siguiente. Es preferible llamar desde la aplicación de visual basic a la base de datos que en el propio fichero php. La conexión es más cómoda para la aplicación.

```
Set BD = New adodb.Connection

cadena_conexion = "driver={MySQL ODBC 5.1 Driver};" & _
"Server=localhost;" & _
"Port=3306;" & _
"Database=bd_lecturas;" & _
"User=root;" & _
"PWD=rootadmin"

BD.Open cadena_conexion

On Error Resume Next
```

Ilustración 8: Ejemplo cadena de conexión base de datos

En ella se puede observar que se crea una cadena de conexión que después se enviara al objeto adodb.Connection para acceder a la base de datos.

Se especifican el nombre de la base de datos o el driver que conlleva su complemento, el servidor web (localhost, la propia maquina donde se ejecuta), el puerto de conexión, el usuario y contraseña de la misma. Si existiera algún problema de conexión se mostrará un mensaje con dicho error.

Cada botón llama otra pantalla. Cada una de estas recoge información necesaria para cada ejecución del fichero php. Como la fecha de fichaje de un operario o el mismo nombre.

Todas las pantallas reflejan el mismo arquetipo. Siguen un orden de tabulación y están diseñadas bajo el mismo patrón. Hará el trabajo del usuario más amigable. Además siguen un orden lógico de ejecución, para que el usuario no se despiste y se pierda en la aplicación.

3.3.2.2 Trabajo de PHP y Gráficos

El código PHP nos ayuda a tratar los ficheros de datos. Los ficheros son de texto plano y se deben obtener los datos exactos para la base de datos.

El administrador cuando accede a su zona y ejecuta la importación de los nuevos datos, lo que está haciendo es ejecutar el fichero correspondiente de lectura. Con los parámetros oportunos introducidos en el sistema, el fichero es capaz de abrir los ficheros, separar sus campos, insertarlos en la base de datos y mostrarlos en pantalla.

Para la lectura de los ficheros logs se necesita un fichero incluido en el código de tipo php denominado “campos.log”. Este fichero identificará cada uno de los campos correspondientes de las tablas de la base de datos. Es fundamental para el correcto funcionamiento de la aplicación.

```
<?php

$num_campos=3;

echo 'entra en campos.php';

$f=$array[1]; /*fecha*/

print"<br>";

echo $array [1];

$h=$array[2]; /*hora*/

echo $array[2];

$t=$array[3];/*tipo*/

echo $array[3];

$aux=" ";

print"<br/>";

/* "i" marcar la posicion final de la cadena*/

for ($y=4;$y<=$i;$y++){

    $aux.=$array[$y]."\n";

}
```

```

echo $aux;

print "<br>";

echo 'SALE DE CAMPOS<br>';

?>

```

Ilustración 8: Código de fichero campos.php

La rutina principal de lectura es común a todos los ficheros. Se basa en obtener una línea tras otra hasta el final del fichero y separar los datos por palabras o concatenaciones de las mismas. Un vez realizado esto y tras pasar por el fichero campos se inserta en la base de datos.

```

$i=0;

while (!feof ($file_apa)) /*lee el fichero php_error*/
{
    $line = fgets ($file_apa, 1024);
    print" $line\n";
    $count++;
    echo" <br><br> ";
    $tok = strtok ($line,$CFG_SEPARADOR1);
    $i=0;
    while ($tok !== false) {
        echo "<font color=blue>Palabra= " . $tok
        . "</font>";
        print " , ";
        $i=$i+1;
        $array[$i]=$tok;
        echo "array[$i]=".$array[$i];
        $tok = strtok($CFG_SEPARADOR2);
        print "<br/> ";
    }
}

```

```
}  
  
include("campos.php");  
  
$sql_phperror="INSERT INTO tb_php_error (FECHA, HORA,  
TIPO,DESEERROR) VALUES ('$f', '$h', '$t', '$aux')";  
  
if (!mysql_query($sql_phperror,$link))  
{  
    die('Error: ' . mysql_error());  
}  
  
Print"<br/>";  
  
echo"INSERTADO EN LA BASE DE DATOS <br/>";  
  
    echo "Siguiete linea>>";  
  
}
```

Ilustración 9: Rutina principal de lectura

Cada una de las palabras obtenidas se puestran en pantalla a la vez que la frase entera leída. Una vez mostrado todas las palabras se muestra las inserciones que se han realizado a la base de datos. Esto último se consigue con una sentencia SQL dentro del código PHP como se muestra a continuación.

```
/*LA CONSULTA NO LA SACA HASTA QUE NO LEA TODO EL FICHERO DE LINEAS
php_error(limpio).log */

$sql_phperror = "SELECT * FROM tb_php_error";

$result_error = mysql_query($sql_phperror);

echo "_DEBERÍA_SACAR_LA_CONSULTA - -----"
-----";

Print"<br/>";

while ($row=mysql_fetch_array($result_error))
{
Print"<br/>";

echo $row["FECHA"];

echo $row["HORA"];

echo " & comi & ";

echo $row["TIPO"];

echo $row["DESERROR"];

}

echo "</br>FIN DE LA CONSULTA-----<br/>";
```

Ilustración 10: Código de inserción de datos en la base de datos

Se puede decir que es una consulta genérica para estudiar que se ha insertado en la base de datos y que el administrador pueda visualizarlo.

El formato que se muestra en pantalla es el siguiente

```
[21-Mar-2010 13:42:49] PHP Warning: Invalid library maybe not a PHP library ZendDebugger.dll in
Unknown on line 0
```

```
Palabra= [21-Mar-2010] , array[1]=[21-Mar-2010
Palabra= 13:42:49] , array[2]=13:42:49]
Palabra= PHP , array[3]=PHP
Palabra= Warning: , array[4]=Warning:
Palabra= Invalid , array[5]=Invalid
Palabra= library , array[6]=library
Palabra= maybe , array[7]=maybe
Palabra= not , array[8]=not
Palabra= a , array[9]=a
Palabra= PHP , array[10]=PHP
Palabra= library , array[11]=library
Palabra= ZendDebugger.dll , array[12]=ZendDebugger.dll
Palabra= in , array[13]=in
Palabra= Unknown , array[14]=Unknown
Palabra= on , array[15]=on
Palabra= line , array[16]=line
Palabra= 0 , array[17]=0
entra en campos.php
[21-Mar-201013:42:49]PHP
Warning: Invalid library maybe not a PHP library ZendDebugger.dll in Unknown on line 0
SALE DE CAMPOS
```

Ilustración 11: Ejemplo del resultado de la lectura de datos del fichero log

El formato que se muestra en pantalla de las inserciones en la base de datos es la siguiente:

```
[17-Jan-2020:36:10 & comi & PHPWarning: mysql_free_result(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\grafico_linea_nue.php on line 94
[17-Jan-2020:49:09 & comi & PHPWarning: mysql_free_result(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\grafico_linea_nue.php on line 94
[17-Jan-2020:49:24 & comi & PHPWarning: mysql_num_rows(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\grafica_snoopyvb.php on line 14
[17-Jan-2020:56:34 & comi & PHPWarning: mysql_num_rows(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\grafica_snoopyvb.php on line 14
[17-Jan-2020:59:21 & comi & PHPWarning: mysql_num_rows(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\grafica_snoopyvb.php on line 14
[17-Jan-2021:03:17 & comi & PHPWarning: mysql_free_result(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\GRAFICA_DINAMICA.PHP on line 80
[17-Jan-2021:03:17 & comi & PHPWarning: mysql_free_result(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\GRAFICA_DINAMICA.PHP on line 81
[18-Jan-2018:08:14 & comi & PHPWarning: mysql_free_result(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\grafico_linea_nue.php on line 94
[18-Jan-2018:08:32 & comi & PHPWarning: mysql_free_result(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\GRAFICA_CONTROLUSU.php on line 73
[18-Jan-2018:08:32 & comi & PHPWarning: mysql_free_result(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\GRAFICA_CONTROLUSU.php on line 74
[18-Jan-2019:45:46 & comi & PHPWarning: mysql_free_result(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\grafico_linea_nue.php on line 94
[18-Jan-2019:46:56 & comi & PHPWarning: mysql_free_result(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\GRAFICA_CONTROLUSU.php on line 73
[18-Jan-2019:46:56 & comi & PHPWarning: mysql_free_result(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\GRAFICA_CONTROLUSU.php on line 74
[18-Jan-2019:54:53 & comi & PHPWarning: mysql_free_result(): supplied argument is not a valid
MySQL result resource in C:\Program Files\wamp\www\grafico_linea_nue.php on line 94
[29-Jan-2012:24:19 & comi & PHPWarning: PHP Startup: Invalid library (maybe not a PHP library)
ZendDebugger.dll in Unknown on line 0
[03-Feb-2012:28:48 & comi & PHPWarning: PHP Startup: Invalid library (maybe not a PHP library)
ZendDebugger.dll in Unknown on line 0
FIN DE LA CONSULTA-----
```

Ilustración 12: Resultado de la inserción en la base de datos

Una vez lanzado este programa ya se puede obtener información de la base de datos. Con esta información, se obtienen los gráficos oportunos. Cada uno de los gráficos está diseñado en código PHP y se valen de las librerías gratuitas graficas existentes en el mercado como “JPGraph”.

El formato de este fichero es común a todos ellos. Bajo una consulta a la base de datos a la tabla concreta obtiene los datos para dibujarlos posteriormente.

El código llama a la librería para mostrar el grafico y a la línea que representa los datos. Los formatos vienen predefinidos por la propia librería pero se modifican para mostrar cómo se quiere en el proyecto. Las abscisas y las leyendas pertinentes que se incluyen, y su formato grafico se muestra en las siguientes líneas.

```
$sql_rango=$hf-$hi;

for ($i=0 ; $i<$sql_rango; $i++)
{
    $sql_phperror[$i]= mysql_query("SELECT ind FROM tb_php_snoopy WHERE
comentario='$usu' AND hora>= '$hi' and hora<= '$hf'");

    $sql_total[$i]=mysql_num_rows($sql_phperror[$i]);

    $hi=$hi+1;

    $ydata[$i]= $sql_total[$i];
}

// Create the graph. These two calls are always required

$graph = new Graph(450,250,"auto");

$graph->SetScale("textlin");

$graph->img->SetAntiAliasing();

$graph->xgrid->Show();

// Create the linear plot

$lineplot=new LinePlot($ydata);

$lineplot->SetColor("red");
```

```
$lineplot->SetWeight(8);  
$lineplot->SetLegend("Evolucion");  
  
// Setup margin and titles  
$graph->img->SetMargin(50,20,20,40);  
$graph->title->Set("ERRORES TIPO PHP POR HORAS");  
$graph->xaxis->title->Set("HORAS");  
$graph->yaxis->title->Set("CASOS");  
$graph->ygrid->SetFill(true,'#EFEFEF@0.5','#F9BB64@0.5');  
// $graph->SetShadow();  
// Add the plot to the graph  
$graph->Add($lineplot);  
// $graph->Add($lineplot2);  
// Display the graph  
$graph->Stroke();
```

Ilustración 13: Código PHP genérico del grafico

Con las librerías JpGraph es posible realizar gráficos a partir de un array de datos, no desde una consulta, los datos obtenidos de la consulta se deben almacenar en una variable de tipo array para poder trabajar con ellos.

Capítulo 4

Planificación y Presupuesto

En este capítulo se mostrarán los datos obtenidos para la planificación y presupuesto del proyecto. La planificación se ha llevado a cabo mediante las herramientas de Microsoft Project, en cambio, para el presupuesto se han utilizado las plantillas proporcionadas por la Universidad Carlos III.

4.1 Planificación

4.1.1 Modelo de gestión del proyecto

El ciclo de vida ha marcado la metodología de trabajo, de principio a fin, del proyecto y la planificación del mismo. El modelo de desarrollo utilizado para la realización de la aplicación ha sido el modelo en espiral (Piattini M.et l, 2003).

El ciclo de vida del desarrollo software es una descripción de las distintas formas de desarrollo de un proyecto, es decir, la orientación que debe seguirse para obtener, a partir de los requisitos del cliente, sistemas que puedan ser utilizados por éste. Este término también se conoce como conjunto de fases, procesos y actividades requeridas para ofertar, desarrollar, probar, integrar, explotar y mantener un producto software.

El ciclo de vida utilizado para el desarrollo define el orden para las tareas o actividades involucradas, también definen la coordinación entre ellas, enlace y realimentación entre las etapas.

El modelo en espiral fue propuesto por Boehm en 1988. Básicamente consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro. En cada una de las iteraciones hay que tener en cuenta las necesidades que debe cubrir el producto así como las diferentes maneras de conseguir los objetivos de forma exitosa. Si el resultado no es el adecuado o se necesita mejorar se planificarán los siguientes pasos y se comienza un nuevo ciclo de la espiral.

Se ha comentado que el ciclo de vida define el orden para las tareas, por tanto vamos a definir las tareas del ciclo de vida en espiral, para cada ciclo habrá cuatro tareas:

1. Fijar los objetivos, fijar las restricciones e identificar los riesgos del proyecto y las estrategias alternativas para evitarlos.
2. Analizar los riesgos posibles que pueden aparecer en el desarrollo.
3. Desarrollar, verificar y validar las tareas.
4. Planificar el desarrollo. Se revisa todo lo hecho y se decide el modo de continuar y planificar la siguiente actividad.

4.1.2 Diagrama de Gantt

Una vez explicados los aspectos generales de la gestión del proyecto, procedemos a detallar la planificación del mismo. Para realizar la planificación se ha utilizado una herramienta básica en la rama de ingeniería del software; el diagrama de Gantt.

El diagrama de Gantt es una popular herramienta gráfica utilizada para mostrar el tiempo previsto de esfuerzo para realizar un trabajo. A pesar de que, en principio, el diagrama de Gantt no indica las relaciones existentes entre actividades, la posición de cada tarea a lo largo del tiempo hace que se puedan identificar dichas relaciones e independencias. Fue Henry Laurence Gantt quien, entre 1910 y 1915, desarrolló y popularizó este tipo de diagrama en Occidente.

En gestión de proyectos, el diagrama de Gantt se ha convertido en una herramienta básica con la finalidad de representar las diferentes unidades mínimas de trabajo y las fases, tareas y actividades programadas como parte de un proyecto.

A groso modo, el diagrama está compuesto por un eje vertical donde se establecen las actividades que constituyen el trabajo que se va a ejecutar, y un eje horizontal que muestra en un calendario, la duración de cada una de ellas.

Las desviaciones en el calendario han sido habituales a lo largo del periodo de desarrollo. Estas demoras son debidas a la necesidad de compaginar el desarrollo del proyecto con el trabajo diario de jornada partida y por tanto, la ralentización y pausas en el desarrollo del trabajo, que no ha así se han tenido todas en cuenta puesto que se ha alargaría demasiado en el tiempo el diagrama.

En la siguiente página se muestra la figura que contiene la planificación del proyecto completo. Teniendo en cuenta que cada día de trabajo corresponden a dos horas reales, excepto en la fase de documentación que corresponde a una hora de trabajo.

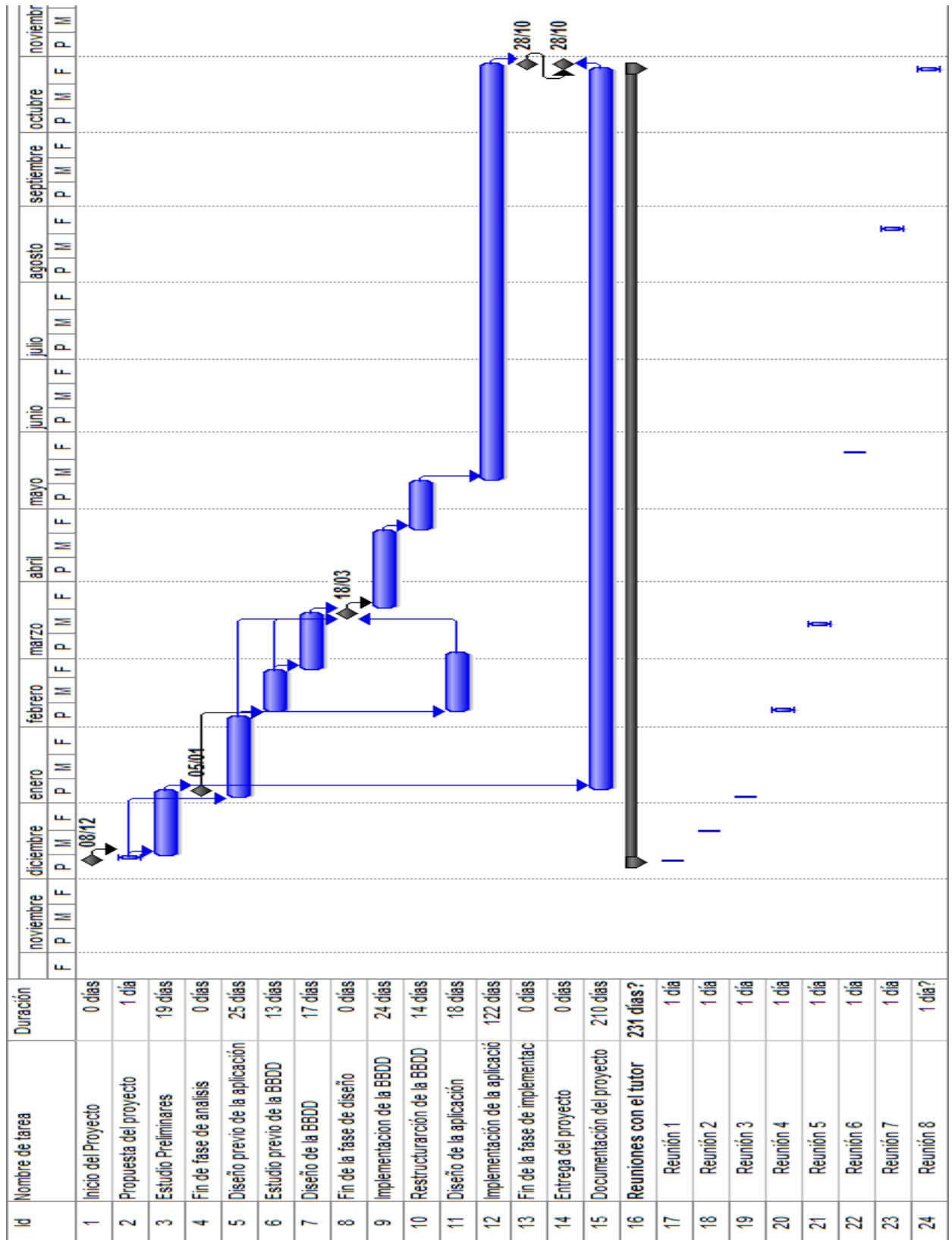


Ilustración 14: Diagrama de Gantt

Según el diagrama de Gantt , mostrado en la figura anterior, podemos decir que el proyecto realizado ha tenido una duración de 233 días, 1.852 horas, donde se han realizado las fases de diseño, análisis, implementación y documentación, además, de algunas reuniones presenciales con el tutor del proyecto.

Como aspectos a destacar, resaltar las diferentes tareas de diseño que han sido realizadas en paralelo para reducir tiempo y recursos.

La tarea que ha durado más tiempo ha sido la de documentación puesto que se comenzó con el hito de 'inicio de proyecto' y finaliza al terminar el proyecto. Esto es debido a que en esta fase se recopila la información necesaria para el desarrollo de los objetivos y se van almacenando datos de interés durante todo el tiempo que ha durado la realización.

El calendario laboral utilizado para la planificación es un calendario estándar donde los periodos laborables se encuentran de lunes a viernes y los periodos de descanso son los sábados y domingos.

4.2 Presupuesto

En este apartado se va a mostrar de manera detallada el presupuesto desglosado del proyecto, especificando los diferentes gastos para su realización.

4.2.1 Resumen de horas dedicadas

Basándonos en el diagrama de Gantt se pueden desglosar exactamente las horas utilizadas en cada fase del proyecto.

Fase de Análisis: 20 días x 4 horas= 80 horas

Fase de Diseño: 55 días x 4 horas= 220 horas

Fase de Implementación: 178 días x 4 horas= 712 horas

Fase de Documentación: 210 días x 4 horas= 840 horas

Por tanto el número de horas totales dedicadas al proyecto será la suma de las horas dedicadas a cada una de las fases. El coste en horas de la totalidad del proyecto es de 1.852 horas.

4.2.2 Resumen de personal

En la siguiente tabla se muestran los cargos correspondientes al personal informático cualificado para realizar las distintas tareas o actividades. Los salarios por hora están en consonancia con los salarios de empleados en empresas similares del sector. Todos los costes son calculados sin I.V.A.

DESCRIPCION	NUMERO DE HORAS	COSTE HORA	DEDICACION (hombre/mes)	TOTAL (€)
Diseñador	80	26 €	1	2.080
Ingeniero Sénior	220	35,72 €	2.75	7.858,4
Ingeniero	712	21,5 €	8.9	15.308
Responsable de documentación	840	16 €	10.5	13.440
TOTAL			23.393,708	

Tabla 42: Resumen de personal

Dedicación de 1 hombre/mes= 80 horas según la planificación realizada.

4.2 Resumen de Hardware

En la siguiente tabla se muestran los equipos informáticos adquiridos con su coste de amortización durante el periodo que dura el proyecto. Todos los costes son calculados sin I.V.A.

DESCRIPCION	UNIDADES	COSTE (€)	COSTE TOTAL (€)	COSTE IMPUTABLE(€)
Ordenador NP-R69L Intel® Core2	1	649	649	154,44
Impresora Xerox® Phaser 8560 Color	1	200	200	38,3
Sticks USB 4 GB	1	12,99	12,99	4,8
Sensor Digital Nitren	1	700	700	26,7
Disco Externo 120" GB	1	160	160	102
TOTAL				326,24

Tabla 43: Resumen de Hardware

Siendo la amortización:

$$\frac{A * C * D}{B}$$

A = nº de meses desde la fecha de facturación

B = periodo de depreciación ()

C = coste del equipo

D = % de uso que se dedica al proyecto

4.2.4 Resumen de Software y Licencias

En las siguientes tablas se muestra las herramientas software necesarias para el proyecto. Todos los costes son calculados sin I.V.A.

DESCRIPCION	UNIDADES	COSTE (€)	COSTE TOTAL (€)
Microsoft Office Professional 2007	1	299	299
Microsoft Office Project 2007	1	59	59
Microsoft Visual Basic 6.0	1	450	450
Internet Explorer 9	1	0	0
Editor Notepad++	1	0	0
WampServer	1	0	0
TOTAL			880

Tabla 44: Resumen de Software

4.2.5 Resumen de Material Fungible

En la siguiente tabla se muestra el material fungible que se estima necesario para la realización del proyecto, así como sus costes. En material de escritorio englobamos folios, bolígrafos, gomas, lápices, clips, carpetas, archivadores, recambios, grapadoras y demás material de oficina. Todos los costes son calculados sin I.V.A.

DESCRIPCION	COSTE TOTAL (€)
Material de Escritorio Variado	80
Recambios de Impresora	100
Material de Limpieza Táctil	25
TOTAL	205

Tabla 45: Resumen Material Fungible

4.2.6 Resumen del presupuesto total

En la siguiente tabla se muestra el sumatorio de los totales anteriormente calculados pero sin el I.V.A. incluido. A la suma de los costes le vamos a añadir un diez por ciento en concepto de costes indirectos, lo cual equilibrará los riesgos del proyecto y aquellos otros valores que no se han tenido en cuenta al realizar el presupuesto.

DESCRIPCION	COSTE TOTAL (€)
Equipo de Trabajo	23.393,71
Amortización	326
Subcontratación de tareas	0
Costes de funcionamiento	1.130
Costes Indirectos (10%)	2.444,14
TOTAL	27.293,85

Tabla 46: Resumen presupuesto total

4.2.7 Plantilla Resumen

1.- Autor

Esther Rodríguez Andrés

2.- Departamento

Informática

3.- Descripción del Proyecto

- Título: **Diseño e implementación de una plataforma para la asistencia en el tratamiento de registros asociados a usuarios.**

- Duración (meses): **10, 52**

Tasa de costes indirectos: **10 %**

4.- Presupuesto total del Proyecto (Valores en Euros)

27.293,85 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y Nombre	NIF. (no rellenar solo a título informativo)	Categoría	Dedicación (hombres/mes)(a)	Coste (hombres/mes)	Coste (Euro)
Esther Rodríguez		Diseñador	1	2.080	2080
Esther Rodríguez		Ingeniero Sénior	2,75	7.858,4	21610,6
Esther Rodríguez		Ingeniero	8,9	15.308	136241,2
Esther Rodríguez		Documentación	10,5	13.440	141120

** Utilizado 1 Hombre mes = 80 horas según la planificación realizada

a) 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)

Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

DESCRIPCION	COSTE (€)	% Uso dedicado proyecto	Dedicación (mes)	Periodo de depreciación	Coste imputable (euros)
Ordenador NP-R69L Intel® Core2	649	100	10	60	108,16
Impresora Xerox® Phaser 8560 Color	200	100	10	60	33,33
Sticks USB 4 GB	12,99	100	10	60	2,16
Sensor Digital Nitren	700	100	10	60	116,66
Disco Externo 120" GB	160	100	10	60	26,66

«) Fórmula de cálculo de la Amortización:

A = n° de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

$$A/B \times C \times D$$

SUBCONTRATACIÓN DE TAREAS - NO APLICA

Descripción	Empresa	Coste imputable
Total		0.00

OTROS COSTES DIRECTOS DEL PROYECTO «)

Descripción	Empresa	Coste imputable
Microsoft Office Professional 2007		299
Microsoft Office Project 2007		59
Microsoft Visual Basic 6.0		450
Material fungible		205
Total		1.013

«) Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes

Presupuestos Costes Totales	Presupuestos Costes Totales
Personal	23.393,71
Amortización	326
Subcontratación de tareas	0
Costes de funcionamiento	1130
Costes indirectos	2444,14
Total	27.293,85

Capítulo 5

Conclusiones y trabajos futuros

En este apartado se exponen las conclusiones que se han obtenido del trabajo realizado, valorando los objetivos cumplidos que se plantearon al principio del proyecto. También se analizan los posibles trabajos futuros que se podrían llevar a cabo para mejorar este proyecto. En último lugar se mostrará la bibliografía utilizada para poder realizar este trabajo.

5.1 Conclusiones

PROCESO

El proyecto comenzó con la idea de mostrar una serie de gráficos representando un eje de abscisas en una aplicación de internet, primeramente se pensó en una aplicación que corriera íntegramente en el explorador de internet, más tarde se decidió en crear una aplicación visual para ayudarnos a gestionarlo.

Se ha realizado un trabajo intenso en cuanto al estudio de documentación acerca del lenguaje PHP y la manera de conseguir alcanzar los objetivos propuestos mediante esta herramienta. Tras un primer acercamiento a este lenguaje me di cuenta que no era familiar para mí y que su dificultad se agravaba al no tener un buen compilador del mismo. En el transcurso de este trabajo comprendí las numerosas posibilidades que existen para tratar variopintos problemas, pero también de algunas limitaciones inherentes a la propia herramienta que no permitían hacer lo deseado en el proyecto.

La principal limitación fue la de mostrar los gráficos en ventana, pues sólo se permitía código PHP. De este modo, se inició un nuevo estudio sobre tecnologías de mercado mediante llamadas en PHP, hasta que se encontró JGraph, que hizo posible seguir con nuestra labor de dibujar gráficos en las ventanas del explorador de internet.

Por otro lado el uso de la aplicación Visual Basic era para mí muy familiar y experimentado por lo que me ayudó mucho a enlazar una aplicación visual creada con esta aplicación y el explorador de internet para mostrar los resultados en formato de eje de abscisas.

Podríamos decir que han sido las principales dificultades a resolver durante el desarrollo del proyecto, sin mencionar la primera de todas, en la fase de análisis, cuando se vio la necesidad de recurrir a un software que realizara las funciones de servidor para poder unir la base de datos con la aplicación.

PERSONAL

Todos estos inconvenientes encontrados por el camino han sido resueltos, creo, que de manera positiva y satisfactoria, pues se han cumplido los objetivos propuestos al inicio del proyecto además del valor añadido que supone haber aprendido a usar tecnologías que me resultaban totalmente desconocidas y que actualmente tienen mucha relevancia como son PHP, SQL y el servidor apache WAMP.

Así mismo, los conocimientos ya adquiridos durante el transcurso de la carrera me han ayudado a resolver ciertas etapas del proyecto de forma satisfactoria. Por ejemplo, el conocimiento de las normas de los esquemas de entidad/relación y el Modelo Relacional me han ayudado a componer un orden lógico de las ideas. El diagrama de Gantt me ha servido para distribuir el proyecto en el tiempo, algo que para mí era un poco difícil de valorar. Así mismo, el conocimiento de las bases de datos y todas sus herramientas en general me han ayudado a organizar la información y tratarla adecuadamente.

PRODUCTO

Para concluir, valoraré los objetivos iniciales expuestos con anterioridad:

- Diseño y desarrollo de una aplicación que muestre de forma clara la información en tiempo real de una situación concreta.

La aplicación diseñada (S.A.U. – Sistema de Administración al Usuario), engloba este objetivo, mostrando claramente la información de una situación concreta mediante un gráfico detallado.

- Estudio de las posibilidades que ofrece las nuevas tecnologías para insertar estas en nuestro proyecto, haciendo las más potentes y amigables de manejar.

Se ha seleccionado las herramientas actuales en el mercado para realizar nuestra labor. Los servidores apache actuales engloban ciertas herramientas (Servidor de extensiones, soporte para diferente código de programación y bases de datos relacionales) que benefician enormemente la labor del programador.

- Búsqueda de soluciones a las limitaciones de desarrollo de las propias herramientas.

En las ocasiones que se han encontrado limitaciones de desarrollo de las propias herramientas se han realizado inserciones de objetos compatibles con las mismas para solucionar esta situación. Por ejemplo, en Visual Basic encontré la problemática de llamar a un objeto de Mozilla Firefox, lo solucioné forzando a que el explorador debiera ser el Internet Explorer 7.0 en adelante, más Genérico y compatible.

- Implantación de una base de datos estable que sea capaz de almacenar los datos de forma correcta y ordenada además de ayudar a un trato fácil de la operación de mantenimiento de la propia base de datos.

El Sistema de base de datos incorporado al servidor apache utilizado (MySQL) cumple con los requisitos anteriormente expuestos. Recordando que un sistema local de este programa aumentaría la seguridad y potencialidad del sistema.

Trabjará de forma opaca al usuario y se comportará como se requiere para el correcto funcionamiento de la aplicación.

- Realización de gráficos usando los datos almacenados en la base de datos.

Muestreo de los datos en un sistema grafico, se ha cumplido. Y con ello la dificultad de sincronizar los datos mostrados en el mismo con la veracidad de los almacenados en la base de datos

Mediante la librería JGraph y el lenguaje de programación PHP se ha podido realizar tal objetivo.

5.2 Trabajos futuros

Teniendo en cuenta que el objetivo principal del proyecto es mostrar graficas que representen la conglomeración de información dada, nuestra aplicación es un prototipo de posibles soluciones que se pueden llevar a cabo para solucionar este problema, por tanto, a partir de este proyecto se pueden realizar mejoras para aumentar las funcionalidades.

Base de datos:

- Almacenar la base de datos en un servidor web, para centralizar los datos y conseguir acceso a ellos desde distintos dispositivos. O bien centralizarlos bajo un sistema local mucho más potente y seguro.
- Implementar la base de datos a partir de otros sistemas gestores para poder realizar comparativas de accesibilidad, fiabilidad y eficacia.

Aplicación web:

- Mejorar la accesibilidad del portal web para aquellos usuarios con algún tipo de discapacidad.
- Aumentar la aplicación de manera de que a partir de cada grafico se pudiera obtener datos de las principales características de cada punto que la compone.

Otras mejoras:

- Desarrollar una nueva aplicación a partir de esta para que pudiera trabajar en tiempo real, esto es, que una vez se recibieran los datos de forma remota la aplicación fuera capaz de optimizar y almacenar los datos para obtener una nueva gráfica, si que el usuario o el administrador deba cargarlos previamente.
- Vincular la aplicación con otra que pudiera obtener más datos a los que saca de la aplicación inicial. Por ejemplo, obtener la productividad y así el salario de un operario en función de esta productividad o retraso horario.

BIBLIOGRAFÍA

Barrionacal, Luis (2008) Introducción a los sitios web dinámicos. *Observatorio Tecnológico- Ministerio de Educación, Política Social y Deporte- Gobierno de España.*

Departamento de base de datos UC3M. Material didáctico de Diseño de Bases de Datos. Conocimientos y técnicas avanzadas para el diseño de bases de datos y acceso a las mismas.

<http://labda.inf.uc3m.es/doku.php>

Introducción a los lenguajes de programación y herramientas. Información general.

<http://es.wikipedia.org>

www.lawebdelprogramador.com

Piattini, José A. Calvo-Manzano, Joaquín Cervera, Luis Fernandez (2003).

Aplicaciones informáticas de gestión. Ingeniería del Software. Modelo en espiral.

Tutoriales de Visual Basic. Programación.

Visual Basic en MSDN (Visual Basic Resource Center).

[http://msdn.microsoft.com/library/keh1dz1\(VS.90\).aspx](http://msdn.microsoft.com/library/keh1dz1(VS.90).aspx)

Microsoft Visual Basic 6.0 Common Controls (Español).

<http://www.microsoft.com/download/en/details.aspx?id=10019>

Tutoriales de JGraph. Programación

Jpgraph.net/

blog.unijime.net/JGraph-gráficos-con-php

<http://www.desarrolloweb.com/articulos/1987.php>

Tutoriales de PHP. Programación

<http://www.programacion.com/pho/tutorial/php/>

<http://www.php.net/manual/es/book.mysql.php>

Tutoriales de SQL. Programación

www.mysql.com

dev.mysql.com/doc/refman/5.0/es/index.html

WorldWideWeb (W3C)-Oficina Española. 2008. World Wide Web

Consortium (W3C)

<http://www.w3c.es>

WAMP Portable. Website Server

<http://portableapps.com/apps/development/wamp>

WAMP Apache Friends. Sitio web acerca de la instalacion y uso del paquete.

<http://www.wampserver.com/en>

<http://www.educasitio.com/wampserver>

ANEXOS

En este último capítulo se anexan ejemplos explicativos del desarrollo y las herramientas de las distintas aplicaciones utilizadas a modo aclarativo.

Anexo 1: Manual de instalación y uso del servidor

1.1 Sobre este manual

Este es el manual de referencia para el servidor de base de datos MySQL, versión 5.0. No está destinado para usarse con versiones más antiguas del software MySQL debido a las numerosas diferencias funcionales y de otro tipo entre MySQL 5.0 y versiones previas.

Este manual es una referencia, por lo que no proporciona instrucciones sobre conceptos generales de SQL o de bases de datos relacionales. Tampoco enseña sobre cómo usar su sistema operativo o su intérprete de línea de comandos.

1.2 Información general

El software MySQL[®] proporciona un servidor de base de datos SQL muy rápido. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido. MySQL es una marca registrada de MySQL AB.

El software MySQL tiene una doble licencia. Los usuarios pueden elegir entre usar el software MySQL como un producto Open Source bajo los términos de la licencia GNU o pueden adquirir una licencia comercial estándar de MySQL AB.

1.3 Instalación

El primer paso es descargar el paquete de aplicaciones XAMPP en su versión portable, cuyo enlace de descarga es el siguiente:

<http://portableapps.com/apps/development/xampp>

Si se prefiere instalar la versión convencional de la aplicación, hay que recurrir al portal oficial de XAMPP

(<http://www.apachefriends.org/es/xampp.html>).

Para instalar esta versión no será necesario seguir los pasos de este apartado.

Cuando el archivo se haya descargado, se deberá ejecutar y se instalará de manera automática con el asistente de instalación convencional. El primer paso es arrancar el servidor, para ello hay que dirigirse a la carpeta creada tras la instalación y ejecutar el archivo "*setup_xampp.bat*". Este archivo cambia la unidad asignada al servidor para que coincida con la que tiene asignado el dispositivo portátil en el que se haya instalado. Posteriormente se arranca el servidor ejecutando el archivo "*xampp_start.exe*".

Si todo este proceso ha ido correctamente, el servidor ya está funcionando. Ahora diríjase al navegador Web y teclee la siguiente dirección:

<http://localhost/phpmyadmin>

Desde esta dirección se accede a la herramienta que actúa de interfaz con el sistema gestor de bases de datos y gracias a la cual se podrá crear de manera sencilla la base de datos necesaria.

En la página de inicio de la herramienta aparece un cuadro que nos permite crear una base de datos. El cuadro se ha de rellenar como se muestra en la siguiente figura:

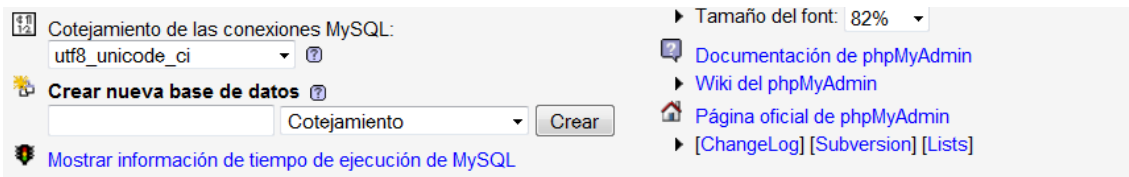


Ilustración 15: Ilustración del modelo de la base de datos

El siguiente paso es asignarle una contraseña al usuario "root" de esta base de datos. Para ello hay que seleccionar la base de datos que se acaba de crear y dirigirse a la sección *privilegios*. En esta sección se editarán los datos del usuario y se le añadirá una contraseña.

Como último paso hay que modificar el fichero de configuración de phpMyAdmin para añadir el cambio de contraseña, si no, phpMyAdmin no funcionará. Para ello hay que acceder al archivo situado en la ruta (*/xampp/phpMyAdmin/config.inc.php*).

En este momento la instalación ha finalizado. A partir de este momento se podrá crear la base de datos y ejecutar desde phpMyAdmin.

Anexo 2: Acerca de la base de datos

2.1 Creación de tablas

Este anexo está dedicado al script SQL de creación de bases de datos. Después de haber explicado las peculiaridades de la implementación, se anexa el script completo para resolver dudas o curiosidades en caso que fuera necesario.

```
CREATE TABLE `bd_lecturas`.`apacheerror` (  
  `ID` INT(11) NOT NULL,  
  `DIASEM` VARCHAR(5) NOT NULL,  
  `MES` VARCHAR(7) NOT NULL,  
  `DIA` VARCHAR(7) NOT NULL,  
  `HORA` TIME NOT NULL,  
  `ANNO` YEAR(4) NOT NULL,  
  `TIPO` TEXT NOT NULL,  
  `COMMENTARIO` TEXT NOT NULL,  
  PRIMARY KEY (`ID`),  
  ) ENGINE = InnoDB;
```

```
CREATE TABLE `pfc`.`bd_lecturas.tb_datoslecturas` (  
  `Nombre` VARCHAR(30) NOT NULL,  
  `ape1` VARCHAR(30) NOT NULL,  
  `ape2` VARCHAR(30) NOT NULL,  
  `fecha` INT(8) NULL,  
  `password` VARCHAR(20) NOT NULL,  
  `conectado` INT(1) NOT NULL,  
  PRIMARY KEY (`Nombre`, `ape1`, `ape2`),  
  ON DELETE CASCADE ON UPDATE CASCADE  
  ) ENGINE = InnoDB;
```

```
CREATE TABLE `bd_lecturas`.`tb_fichajes` (  
  `logindex` INT(11) NOT NULL UNIQUE,  
  `logtime` TIMESTAMP NOT NULL,  
  `fecha` VARCHAR(10) NOT NULL,  
  `hora` VARCHAR(10) NOT NULL,  
  `userid` VARCHAR(30) NULL,  
  PRIMARY KEY (`logtime`),  
  ON DELETE CASCADE ON UPDATE CASCADE  
  ) ENGINE = InnoDB;
```

```
CREATE TABLE `bd_lecturas`.`tb_ficheros` (
  `id` INT( 20 ) NOT NULL UNIQUE,
  `nombre fichero` VARCHAR(50) NOT NULL ,
  `fichero_log` VARCHAR( 50 ) NOT NULL ,
  `fichero_campos` VARCHAR( 50 ) NOT NULL ,
  `tabla` VARCHAR( 20 ) NOT NULL ,
  PRIMARY KEY ( `ID` ),
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE = InnoDB;
```

```
CREATE TABLE `bd_lecturas`.`tb_php_error` (
  `id` INT( 50 ) NOT NULL UNIQUE,
  `fecha` VARCHAR(10) NOT NULL ,
  `hora` VARCHAR( 8 ) NOT NULL ,
  `tipo` VARCHAR( 8 ) NOT NULL ,
  `deserror` VARCHAR( 1024 ) NOT NULL ,
  PRIMARY KEY ( `ID` ),
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE = InnoDB;
```

```
CREATE TABLE `bd_lecturas`.`tb_php_snoopy` (
  `ind` INT( 20 ) NOT NULL UNIQUE,
  `hora` VARCHAR(10) NOT NULL ,
  `dia` int( 11 ) NOT NULL ,
  `skx` VARCHAR( 10 ) NOT NULL ,
  `comentario` VARCHAR( 20 ) NOT NULL ,
  PRIMARY KEY ( `Ind` ),
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE = InnoDB;
```

2.2 Diccionario de Datos

El diccionario de datos es la lista de todos los elementos que forman parte del flujo de datos de todo el sistema, en este se guardan los detalles y descripción de todos estos elementos.

TABLA APACHEERROR

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>ID</u>	int(11)			No		auto_increment
<input type="checkbox"/>	DIASEM	varchar(5)	latin1_swedish_ci		No		
<input type="checkbox"/>	MES	varchar(7)	latin1_swedish_ci		No		
<input type="checkbox"/>	DIA	varchar(7)	latin1_swedish_ci		No		
<input type="checkbox"/>	HORA	time			No		
<input type="checkbox"/>	ANNO	year(4)			No		
<input type="checkbox"/>	TIPO	text	latin1_swedish_ci		No		
<input type="checkbox"/>	COMENTARIO	text	latin1_swedish_ci		No		

Ilustración 16: Ilustración tabla acreedor

TABLA DATOSLECTURAS

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>NOMBRE</u>	varchar(30)	latin1_swedish_ci		No		
<input type="checkbox"/>	<u>APE1</u>	varchar(30)	latin1_swedish_ci		No		
<input type="checkbox"/>	<u>APE2</u>	varchar(30)	latin1_swedish_ci		No		
<input type="checkbox"/>	FECHA	int(8)			Sí	NULL	
<input type="checkbox"/>	PASSWORD	varchar(20)	latin1_swedish_ci		No		
<input type="checkbox"/>	CONECTADO	int(1)			No		

Ilustración 17: Ilustración tabla datoslecturas

TABLA FICHAJES

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>logindex</u>	int(11)			No		auto_increment
<input type="checkbox"/>	<u>logtime</u>	timestamp			No	CURRENT_TIMESTAMP	
<input type="checkbox"/>	fecha	varchar(10)	utf8_general_ci		No		
<input type="checkbox"/>	hora	varchar(10)	utf8_general_ci		No		
<input type="checkbox"/>	userid	varchar(16)	utf8_general_ci		Sí	NULL	

Ilustración 18: Ilustración tabla fichajes

TABLA FICHEROS

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	id	int(20)			No		auto_increment
<input type="checkbox"/>	nombre_fichero	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	fichero_log	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	fichero_campos	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	tabla	varchar(20)	latin1_swedish_ci		No		

Ilustración 19: Ilustración tabla ficheros

TABLA PHP_ERROR

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	ID	int(50)			No		auto_increment
<input type="checkbox"/>	FECHA	varchar(10)	latin1_swedish_ci		No		
<input type="checkbox"/>	HORA	varchar(8)	latin1_swedish_ci		No		
<input type="checkbox"/>	TIPO	varchar(8)	latin1_swedish_ci		No		
<input type="checkbox"/>	DESERROR	varchar(1024)	latin1_swedish_ci		No		

Ilustración 20: Ilustración tabla php_error

TABLA_PHP_SNOOPY

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	mes	varchar(4)	latin1_swedish_ci		No		
<input type="checkbox"/>	dia	int(11)			No		
<input type="checkbox"/>	hora	varchar(10)	latin1_swedish_ci		No		
<input type="checkbox"/>	skx	varchar(10)	latin1_swedish_ci		No		
<input type="checkbox"/>	comentario	varchar(20)	latin1_swedish_ci		No		
<input type="checkbox"/>	ind	int(20)			No		auto_increment

Ilustración 21: Ilustración tabla acreedor php_snoopy

Anexo 3: Acerca de la aplicación

La aplicación no ha sido creada para su uso por cualquier tipo de usuario, es un tipo de aplicación destinada a organizaciones o empresas que requieran llevar a cabo un proceso de control y de optimización de la información. Por tanto, se ha intentado hacer un diseño adecuado al tipo de usuario en cuestión, facilitando la accesibilidad y el uso de la misma.

La primera pantalla es intuitiva, para que el usuario se posicione según la zona a tratar y sus permisos dentro del sistema:

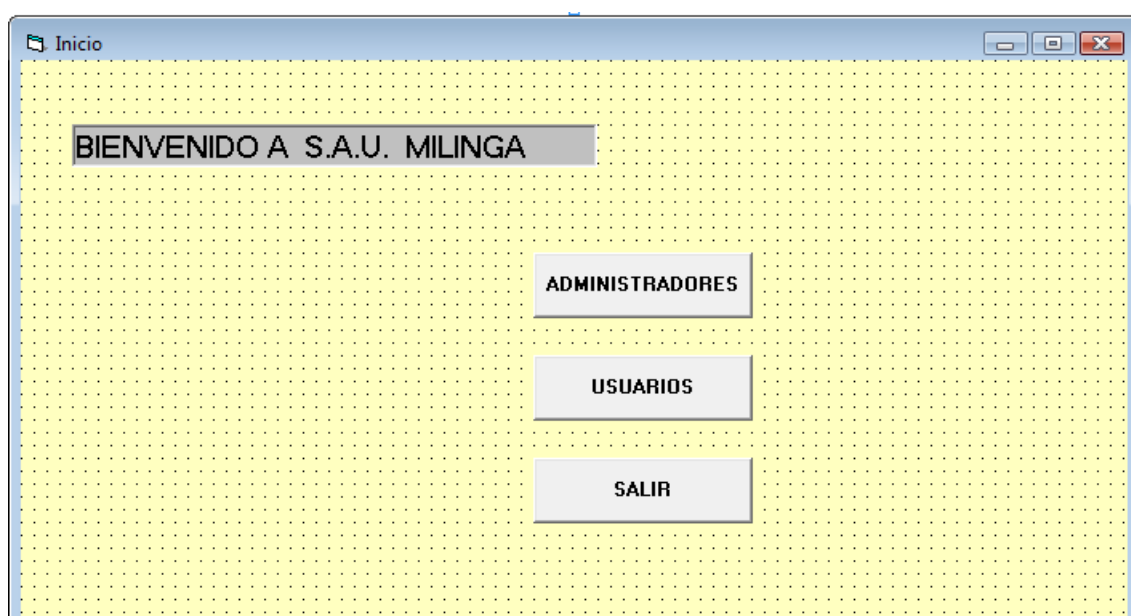
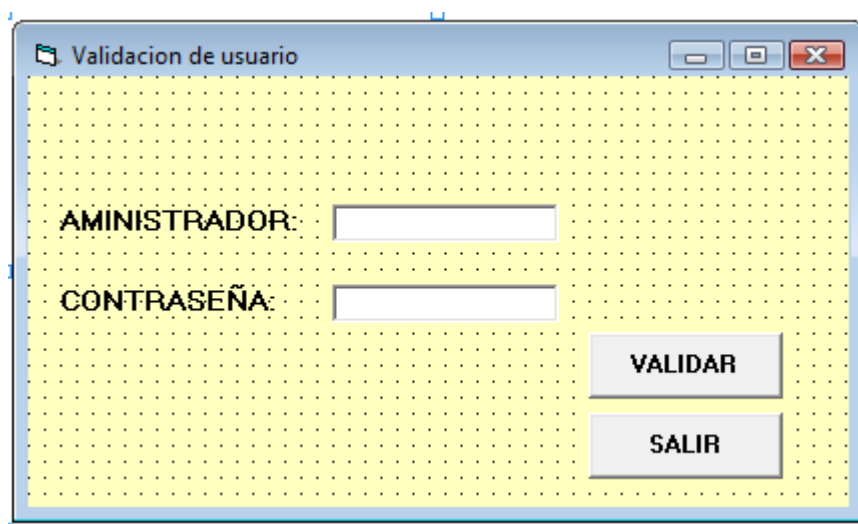


Ilustración 22: Pantalla de inicio

Una vez seleccionado cada de los botones se accede a distintas pantallas que se muestran a continuación.

Si el usuario selecciona el botón de administración, aparece una pantalla intermedia solicitando su usuario y contraseña para acceder a la siguiente pantalla:



Validacion de usuario

AMINISTRADOR:

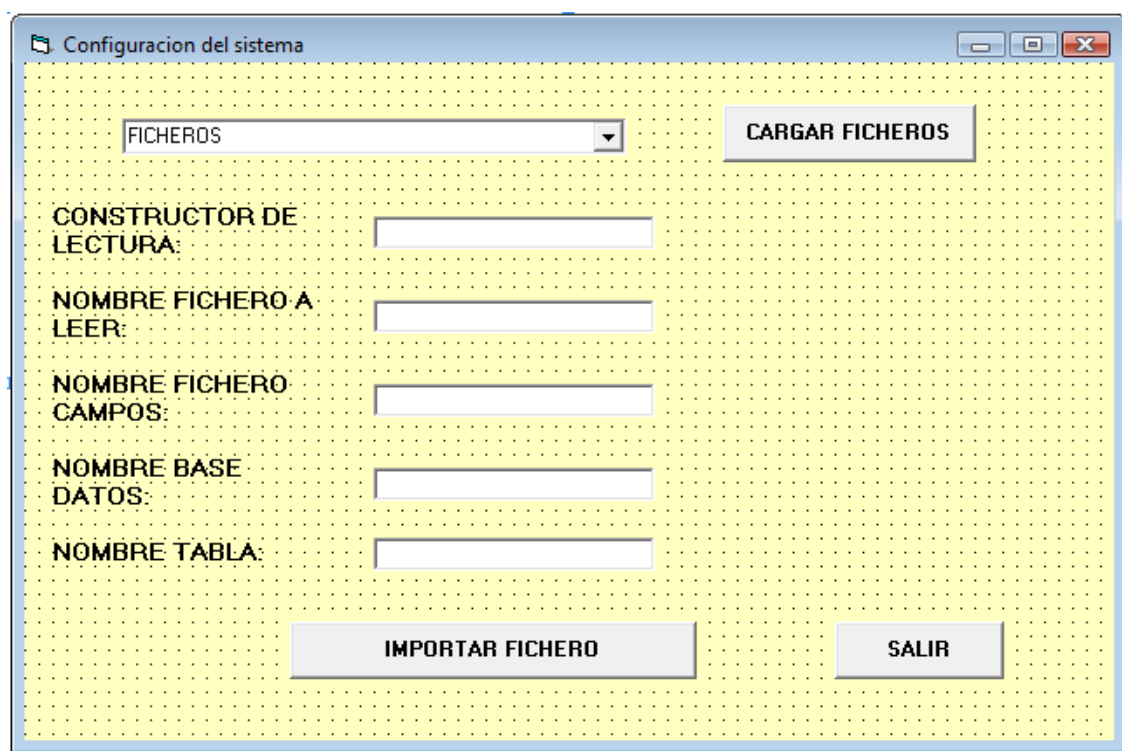
CONTRASEÑA:

VALIDAR

SALIR

Ilustración 23: Pantalla validar usuario

Siendo la validación correcta aparece la siguiente pantalla, donde el administrador puede recargar de nuevos los ficheros o cargar los nuevos:



Configuracion del sistema

FICHEROS

CARGAR FICHEROS

CONSTRUCTOR DE LECTURA:

NOMBRE FICHERO A LEER:

NOMBRE FICHERO CAMPOS:

NOMBRE BASE DATOS:

NOMBRE TABLA:

IMPORTAR FICHERO

SALIR

Ilustración 24: Configurar el sistema

Si por otro lado si se determinó entrar por la parte de usuario la aplicación mostrará la siguiente pantalla, en la que se podrán realizar múltiples acciones:

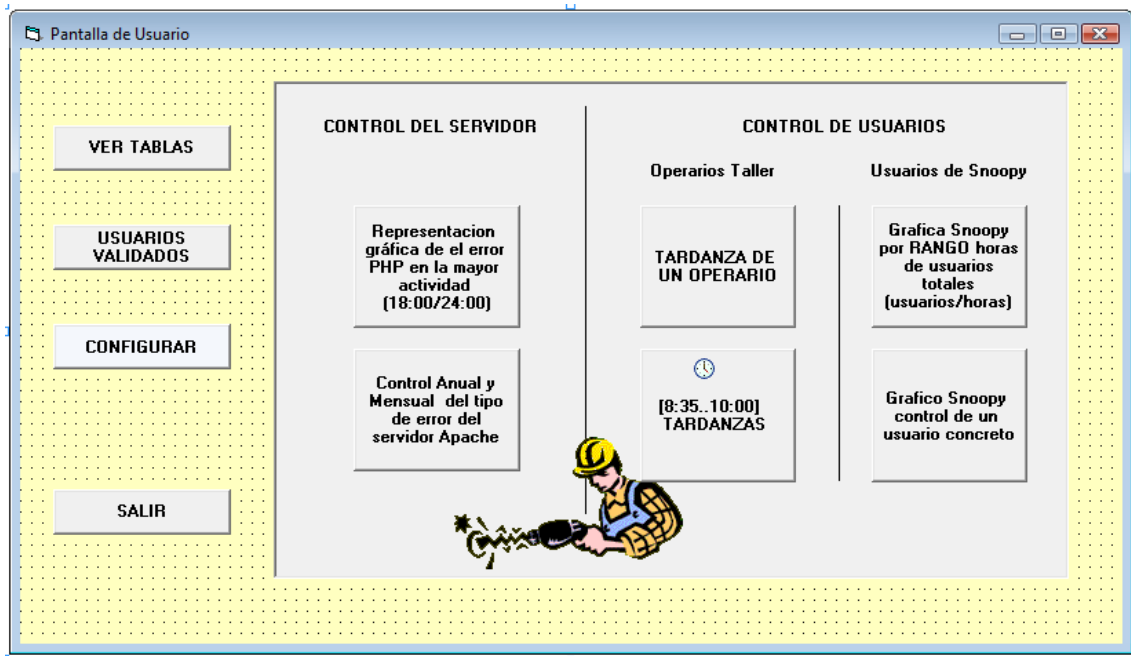
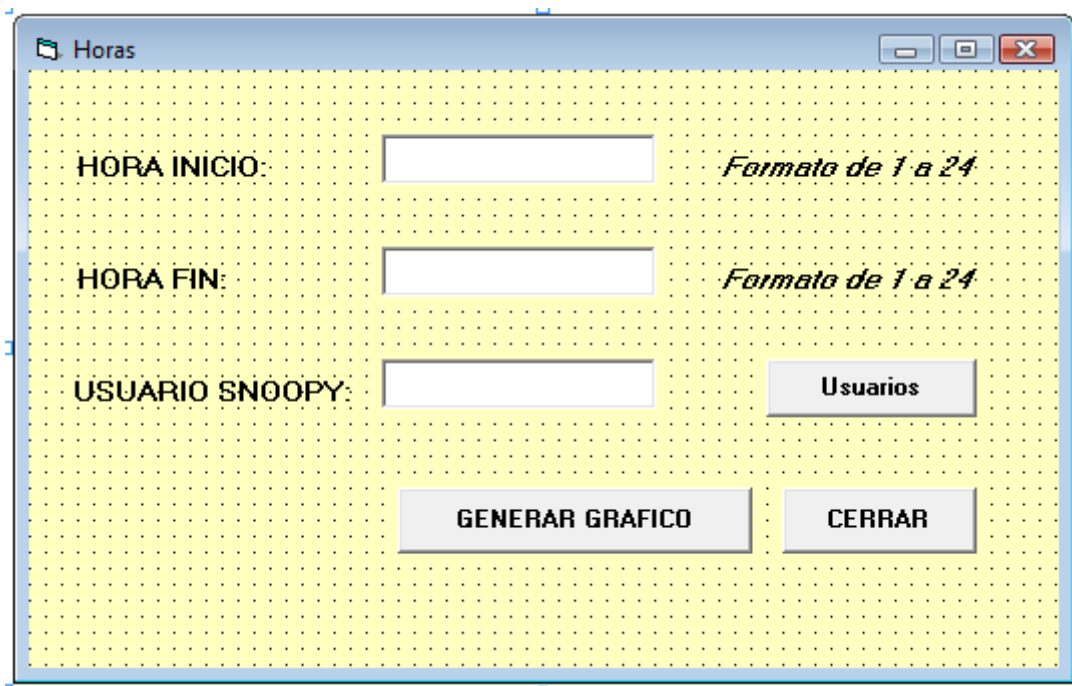


Ilustración 25: Pantalla de usuario

Seleccionando los botones de la parte “Control de usuarios” se accede a una pantalla intermedia de preselección de datos, para que se muestren los datos filtrados según la predilección del usuario.



The image shows a screenshot of a software window titled "Horas". The window has a yellow background with a dotted pattern. It contains three input fields for data selection:

- HORA INICIO:** A text input field with the label "Formato de 1 a 24" to its right.
- HORA FIN:** A text input field with the label "Formato de 1 a 24" to its right.
- USUARIO SNOOPY:** A text input field.

Below the input fields, there are three buttons:

- Usuarios:** A button located to the right of the "USUARIO SNOOPY" field.
- GENERAR GRAFICO:** A button located below the "USUARIO SNOOPY" field.
- CERRAR:** A button located to the right of the "GENERAR GRAFICO" button.

Ilustración 26: Pantalla genérica de preselección de datos

A través de estas pantallas se muestra el gráfico de los datos solicitados en una pantalla intuitiva y refleja de los datos de forma gráfica en eje de cartesiano:

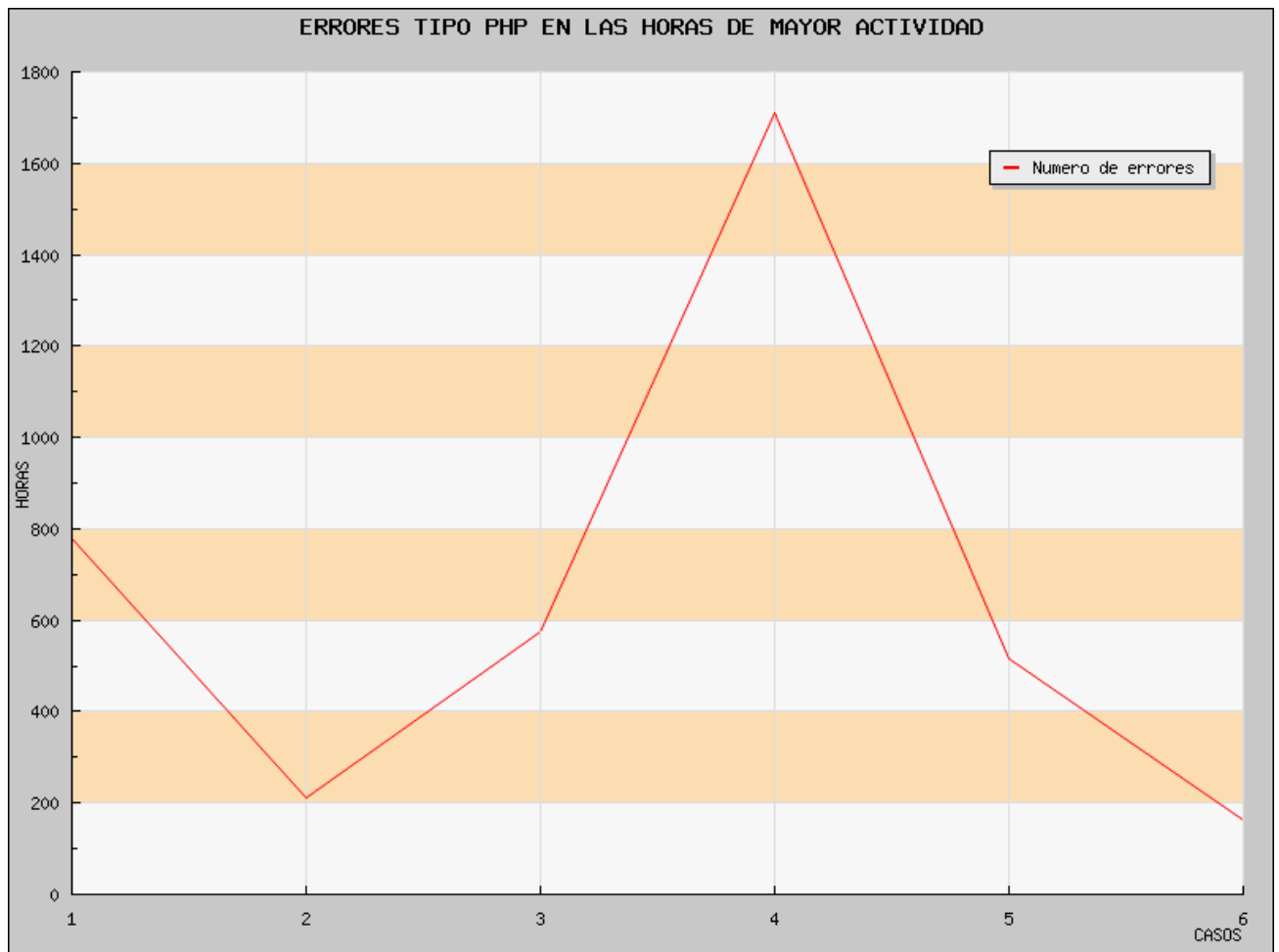


Ilustración 27: Grafico de Numero de errores en el servidor . Grafico genérico.

